Copyright Information

Disclaimer

The following publication, CCIE R&S Lab Workbook Volume I Version 5.0, is designed to assist candidates in the preparation for Cisco Systems' CCIE Routing & Switching Lab Exam. While every effort has been made to ensure that all material is as complete and accurate as possible, the enclosed material is presented on an "as is" basis. Neither the authors nor Internetwork Expert, Inc. assume any liability or responsibility to any person or entity with respect to loss or damages incurred from the information contained in this workbook.

This workbook was developed by Internetwork Expert, Inc. and is an original work of the aforementioned authors. Any similarities between material presented in this workbook and actual CCIE lab material is completely coincidental.

Table of Contents

RIP

> **Note**
>
> Load the *RIP* initial configurations prior to starting. Note that R4's connection to VLAN 146 and the Serial link between R2 and R3 are disabled.

4.1 Basic RIP Configuration

- Configure RIPv2 on all interfaces of all devices in the internal network.
- Disable auto-summary.
- R4 and R6 should be learning RIP routes from BB3 and BB1 respectively.
- Test reachability to all networks and note any problems within the topology.

4.2 RIPv2 Authentication

- Configure RIPv2 authentication on the Ethernet link between R2 and BB2.
- Use the MD5 key number 1 with a password of CISCO.
- R2 should be learning RIP routes from BB2.
- Configure clear-text RIP authentication on the segment between R1 and R6 using the password CCIE.

4.3 RIPv2 Split Horizon

- Disable split-horizon on R5's connection to the Frame Relay cloud.
- Test reachability to all networks and note any changes within the topology.

4.4 RIPv2 Auto-Summary

- Enable auto-summary under the RIP process of R4.
- Note any changes in the network advertisements that R4 is sending.

4.5 RIP Send and Receive Versions

- Remove the **`version 2`** commands under the RIP processes of SW2 and SW4.
- Configure SW2 to send and receive only RIPv2 updates on VLAN 58.
- Configure SW2 to send and receive only RIPv1 updates on the link to SW4.
- Note any changes in reachability or routing information in the network.

4.6 RIPv2 Manual Summarization

- Configure R4 to send two summary routes for the RIP networks learned from BB3 to R5.
- Ensure that these summaries do not overlap any address space that R4 does not have a longer match route to.

4.7 RIPv2 Convergence Timers

- Change the RIP timers throughout the topology to make convergence three times faster than the default.
- Ensure that this configuration does not affect the links to the BB routers.

4.8 RIPv2 Offset List

- Configure an offset-list on R6 so that all traffic going to BB1 uses the Ethernet link to SW1.
- If this link is down traffic should be rerouted over the link to R1.

4.9 RIPv2 Filtering with Passive Interface

- Configure the passive interface feature on SW2 so that it learns RIP updates from SW4 but does not advertise any information back to SW4.

4.10 RIPv2 Filtering with Prefix-Lists

- Configure a prefix-list on R5 so that it does not advertise the two RIP summaries of the BB3 networks to SW2, and so that all other networks are allowed to be advertised.
- Configure a prefix-list on R5 so that it does not install any updates received from R4 on the Frame Relay network. This configuration should not affect the updates received from other neighbors on this segment.
- Ensure that route feedback does not occur for the summaries that R4 is generating to R5.

4.11 RIPv2 Filtering with Standard Access-Lists

- Configure a one line standard access-list on R6 to filter out all routes coming from BB1 that have an even number in the third octet.

4.12 RIPv2 Filtering with Extended Access-Lists

- Remove the previous prefix-list filters on R5.
- Configure an extended access-list filter on R5 so that the routes to VLANs 7 and 9 are only received from R1, while the routes to SW1 and SW3's Loopback0 networks are only received from R3.
- This filter should not affect any other updates on this segment.

4.13 RIPv2 Filtering with Offset Lists

- Configure an offset-list on SW1 so that SW3 does not install a route to VLAN 5.
- This filter should not affect any other updates on this segment.

4.14 RIPv2 Filtering with Administrative Distance

- Configure administrative distance filtering on R5 so that devices within the network cannot reach R4's Loopback0 network.
- This filter should not affect any other networks in the topology.

4.15 RIPv2 Filtering with Per Neighbor AD

- Configure administrative distance filtering on SW1 so that traffic destined for R3's Loopback0 network is sent towards R6.
- This configuration should not affect any other networks in the topology.

4.16 RIPv2 Default Routing

- Configure R6 to advertise a default route to R1 via RIP.
- R6 should not send this default route directly to SW1.
- Do not use any access-list or prefix-lists to accomplish this.

4.17 RIPv2 Conditional Default Routing

- Remove the previous default route advertisement on R6.
- Configure R4 to originate a default route into the RIP domain.
- If the link to BB3 goes down R4 should withdraw its default advertisement.

4.18 RIPv2 Reliable Conditional Default Routing

- Configure the IP SLA feature on R4 to track ICMP reachability to BB3.
- Modify the previous default route origination on R4 so that if an ICMP echo-reply is not received from BB3 R4 withdraws its default advertisement.

4.19 RIPv2 Unicast Updates

- Configure R5 and SW2 so that RIPv2 updates sent over VLAN 58 use unicasts instead of multicasts.

4.20 RIPv2 Broadcast Updates

- Configure R1 and R6 so that RIPv2 updates sent over VLAN 146 use broadcasts instead of multicasts.

4.21 RIPv2 Triggered Updates

- Configure R4 and R5 so that RIPv2 updates are only exchanged over the low speed point-to-point Serial link between them when there is a change in the RIP topology.

4.22 RIPv2 Source Validation

- Configure R1 and R3 to use PPP on the Serial link between them.
- Remove R1's IP address on this segment.
- Configure R3 to assign R1 the IP address 155.1.13.1/32 via IPCP.
- Ensure that RIPv2 updates sent across this link can be installed in the routing tables of these two devices.

RIP Solutions

4.1 Basic RIP Configuration

- Configure RIPv2 on all interfaces of all devices in the internal network.
- Disable auto-summary.
- R4 and R6 should be learning RIP routes from BB3 and BB1 respectively.
- Test reachability to all networks and note any problems within the topology.

Configuration

```
R1:
router rip
 version 2
 network 150.1.0.0
 network 155.1.0.0
 no auto-summary

R2:
router rip
 version 2
 network 150.1.0.0
 network 155.1.0.0
 network 192.10.1.0
 no auto-summary

R3:
router rip
 version 2
 network 150.1.0.0
 network 155.1.0.0
 no auto-summary

R4:
router rip
 version 2
 network 150.1.0.0
 network 155.1.0.0
 network 204.12.1.0
 no auto-summary

R5:
router rip
 version 2
 network 150.1.0.0
 network 155.1.0.0
 no auto-summary
```

```
R6:
router rip
 version 2
 network 54.0.0.0
 network 150.1.0.0
 network 155.1.0.0
 no auto-summary

SW1:
ip routing
!
router rip
 version 2
 network 150.1.0.0
 network 155.1.0.0
 no auto-summary

SW2:
ip routing
!
router rip
 version 2
 network 150.1.0.0
 network 155.1.0.0
 no auto-summary

SW3:
ip routing
!
router rip
 version 2
 network 150.1.0.0
 network 155.1.0.0
 no auto-summary

SW4:
ip routing
!
router rip
 version 2
 network 150.1.0.0
 network 155.1.0.0
 no auto-summary
```

Verification

```
Rack1R5#debug ip rip
RIP protocol debugging is on
```

Note

R5 receives an update from R1 which includes the routes to BB1

```
RIP: received v2 update from 155.1.0.1 on Serial0/0.1
     54.1.1.0/24 via 0.0.0.0 in 2 hops
     150.1.1.0/24 via 0.0.0.0 in 1 hops
     150.1.3.0/24 via 0.0.0.0 in 2 hops
     150.1.6.0/24 via 0.0.0.0 in 2 hops
     150.1.7.0/24 via 0.0.0.0 in 3 hops
     150.1.9.0/24 via 0.0.0.0 in 4 hops
     155.1.7.0/24 via 0.0.0.0 in 3 hops
     155.1.9.0/24 via 0.0.0.0 in 4 hops
     155.1.13.0/24 via 0.0.0.0 in 1 hops
     155.1.37.0/24 via 0.0.0.0 in 2 hops
     155.1.67.0/24 via 0.0.0.0 in 2 hops
     155.1.79.0/24 via 0.0.0.0 in 3 hops
     155.1.146.0/24 via 0.0.0.0 in 1 hops
     212.18.0.0/24 via 0.0.0.0 in 3 hops
     212.18.1.0/24 via 0.0.0.0 in 3 hops
     212.18.2.0/24 via 0.0.0.0 in 3 hops
     212.18.3.0/24 via 0.0.0.0 in 3 hops
```

R5 receives an update from R2 which includes the route to the link to BB2.

```
RIP: received v2 update from 155.1.0.2 on Serial0/0.1
     150.1.2.0/24 via 0.0.0.0 in 1 hops
     192.10.1.0/24 via 0.0.0.0 in 1 hops
```

R5 receives updates from R4 from both the Frame Relay and point-to-point links which include the routes to BB3.

```
RIP: received v2 update from 155.1.0.4 on Serial0/0.1
     30.0.0.0/16 via 0.0.0.0 in 2 hops
     30.1.0.0/16 via 0.0.0.0 in 2 hops
     30.2.0.0/16 via 0.0.0.0 in 2 hops
     30.3.0.0/16 via 0.0.0.0 in 2 hops
     31.0.0.0/16 via 0.0.0.0 in 2 hops
     31.1.0.0/16 via 0.0.0.0 in 2 hops
     31.2.0.0/16 via 0.0.0.0 in 2 hops
     31.3.0.0/16 via 0.0.0.0 in 2 hops
     54.1.1.0/24 via 0.0.0.0 in 4 hops
     150.1.1.0/24 via 0.0.0.0 in 3 hops
     150.1.2.0/24 via 0.0.0.0 in 3 hops
     150.1.3.0/24 via 0.0.0.0 in 3 hops
     150.1.4.0/24 via 0.0.0.0 in 1 hops
     150.1.6.0/24 via 0.0.0.0 in 4 hops
     150.1.7.0/24 via 0.0.0.0 in 4 hops
     150.1.9.0/24 via 0.0.0.0 in 5 hops
     155.1.7.0/24 via 0.0.0.0 in 4 hops
     155.1.9.0/24 via 0.0.0.0 in 5 hops
     155.1.13.0/24 via 0.0.0.0 in 3 hops
     155.1.37.0/24 via 0.0.0.0 in 3 hops
     155.1.45.0/24 via 0.0.0.0 in 1 hops
     155.1.67.0/24 via 0.0.0.0 in 4 hops
     155.1.79.0/24 via 0.0.0.0 in 4 hops
     155.1.146.0/24 via 0.0.0.0 in 3 hops
     192.10.1.0/24 via 0.0.0.0 in 3 hops
RIP: received v2 update from 155.1.0.4 on Serial0/0.1
     204.12.1.0/24 via 0.0.0.0 in 1 hops
     212.18.0.0/24 via 0.0.0.0 in 5 hops
     212.18.1.0/24 via 0.0.0.0 in 5 hops
     212.18.2.0/24 via 0.0.0.0 in 5 hops
     212.18.3.0/24 via 0.0.0.0 in 5 hops

RIP: received v2 update from 155.1.45.4 on Serial0/1
     30.0.0.0/16 via 0.0.0.0 in 2 hops
     30.1.0.0/16 via 0.0.0.0 in 2 hops
     30.2.0.0/16 via 0.0.0.0 in 2 hops
     30.3.0.0/16 via 0.0.0.0 in 2 hops
     31.0.0.0/16 via 0.0.0.0 in 2 hops
     31.1.0.0/16 via 0.0.0.0 in 2 hops
     31.2.0.0/16 via 0.0.0.0 in 2 hops
     31.3.0.0/16 via 0.0.0.0 in 2 hops
     150.1.4.0/24 via 0.0.0.0 in 1 hops
     155.1.0.0/24 via 0.0.0.0 in 1 hops
     204.12.1.0/24 via 0.0.0.0 in 1 hops
```

Due to split horizon being enabled on the multipoint Frame Relay subinterface of R5, the routes coming from R1, R2, R3, and R4 on this link cannot be advertised back out the Frame Relay. However, updates from R1, R2, and R3 on the Frame Relay network can be advertised to R4 out the point-to-point link.

This means that R4, SW2, and SW4 will have full routing information to everyone, per the below output.

```
Rack1R4#show ip route rip
     155.1.0.0/24 is subnetted, 14 subnets
R       155.1.146.0 [120/2] via 155.1.45.5, 00:00:25, Serial0/1
R       155.1.10.0 [120/3] via 155.1.45.5, 00:00:25, Serial0/1
                   [120/3] via 155.1.0.5, 00:00:13, Serial0/0.1
R       155.1.8.0 [120/2] via 155.1.45.5, 00:00:25, Serial0/1
                  [120/2] via 155.1.0.5, 00:00:13, Serial0/0.1
R       155.1.9.0 [120/4] via 155.1.45.5, 00:00:25, Serial0/1
R       155.1.13.0 [120/2] via 155.1.45.5, 00:00:25, Serial0/1
R       155.1.7.0 [120/3] via 155.1.45.5, 00:00:25, Serial0/1
R       155.1.5.0 [120/1] via 155.1.45.5, 00:00:25, Serial0/1
                  [120/1] via 155.1.0.5, 00:00:13, Serial0/0.1
R       155.1.58.0 [120/1] via 155.1.45.5, 00:00:25, Serial0/1
                   [120/1] via 155.1.0.5, 00:00:13, Serial0/0.1
R       155.1.37.0 [120/2] via 155.1.45.5, 00:00:25, Serial0/1
R       155.1.79.0 [120/3] via 155.1.45.5, 00:00:25, Serial0/1
R       155.1.67.0 [120/3] via 155.1.45.5, 00:00:25, Serial0/1
R       155.1.108.0 [120/2] via 155.1.45.5, 00:00:26, Serial0/1
                    [120/2] via 155.1.0.5, 00:00:14, Serial0/0.1
     54.0.0.0/24 is subnetted, 1 subnets
R       54.1.1.0 [120/3] via 155.1.45.5, 00:00:27, Serial0/1
R    212.18.1.0/24 [120/4] via 155.1.45.5, 00:00:26, Serial0/1
R    212.18.0.0/24 [120/4] via 155.1.45.5, 00:00:27, Serial0/1
R    212.18.3.0/24 [120/4] via 155.1.45.5, 00:00:27, Serial0/1
R    212.18.2.0/24 [120/4] via 155.1.45.5, 00:00:27, Serial0/1
R    192.10.1.0/24 [120/2] via 155.1.45.5, 00:00:27, Serial0/1
     31.0.0.0/16 is subnetted, 4 subnets
R       31.3.0.0 [120/1] via 204.12.1.254, 00:00:08, FastEthernet0/0
R       31.2.0.0 [120/1] via 204.12.1.254, 00:00:08, FastEthernet0/0
R       31.1.0.0 [120/1] via 204.12.1.254, 00:00:08, FastEthernet0/0
R       31.0.0.0 [120/1] via 204.12.1.254, 00:00:08, FastEthernet0/0
     150.1.0.0/24 is subnetted, 10 subnets
R       150.1.7.0 [120/3] via 155.1.45.5, 00:00:27, Serial0/1
R       150.1.6.0 [120/3] via 155.1.45.5, 00:00:28, Serial0/1
R       150.1.5.0 [120/1] via 155.1.45.5, 00:00:28, Serial0/1
                  [120/1] via 155.1.0.5, 00:00:15, Serial0/0.1
R       150.1.3.0 [120/2] via 155.1.45.5, 00:00:28, Serial0/1
R       150.1.2.0 [120/2] via 155.1.45.5, 00:00:28, Serial0/1
R       150.1.1.0 [120/2] via 155.1.45.5, 00:00:28, Serial0/1
R       150.1.10.0 [120/3] via 155.1.45.5, 00:00:28, Serial0/1
                   [120/3] via 155.1.0.5, 00:00:15, Serial0/0.1
R       150.1.9.0 [120/4] via 155.1.45.5, 00:00:28, Serial0/1
R       150.1.8.0 [120/2] via 155.1.45.5, 00:00:28, Serial0/1
                  [120/2] via 155.1.0.5, 00:00:15, Serial0/0.1
     30.0.0.0/16 is subnetted, 4 subnets
R       30.2.0.0 [120/1] via 204.12.1.254, 00:00:09, FastEthernet0/0
```

```
R       30.3.0.0 [120/1] via 204.12.1.254, 00:00:09, FastEthernet0/0
R       30.0.0.0 [120/1] via 204.12.1.254, 00:00:09, FastEthernet0/0
R       30.1.0.0 [120/1] via 204.12.1.254, 00:00:09, FastEthernet0/0
```

```
Rack1SW2#show ip route rip
R    204.12.1.0/24 [120/2] via 155.1.58.5, 00:00:09, Vlan58
     155.1.0.0/24 is subnetted, 14 subnets
R       155.1.146.0 [120/2] via 155.1.58.5, 00:00:09, Vlan58
R       155.1.10.0 [120/1] via 155.1.108.10, 00:00:09, Port-channel1
R       155.1.9.0 [120/4] via 155.1.58.5, 00:00:09, Vlan58
R       155.1.13.0 [120/2] via 155.1.58.5, 00:00:09, Vlan58
R       155.1.0.0 [120/1] via 155.1.58.5, 00:00:09, Vlan58
R       155.1.7.0 [120/3] via 155.1.58.5, 00:00:09, Vlan58
R       155.1.5.0 [120/1] via 155.1.58.5, 00:00:09, Vlan58
R       155.1.45.0 [120/1] via 155.1.58.5, 00:00:10, Vlan58
R       155.1.37.0 [120/2] via 155.1.58.5, 00:00:10, Vlan58
R       155.1.79.0 [120/3] via 155.1.58.5, 00:00:10, Vlan58
R       155.1.67.0 [120/3] via 155.1.58.5, 00:00:10, Vlan58
     54.0.0.0/24 is subnetted, 1 subnets
R       54.1.1.0 [120/3] via 155.1.58.5, 00:00:10, Vlan58
R    212.18.1.0/24 [120/4] via 155.1.58.5, 00:00:10, Vlan58
R    212.18.0.0/24 [120/4] via 155.1.58.5, 00:00:10, Vlan58
R    212.18.3.0/24 [120/4] via 155.1.58.5, 00:00:10, Vlan58
R    212.18.2.0/24 [120/4] via 155.1.58.5, 00:00:10, Vlan58
R    192.10.1.0/24 [120/2] via 155.1.58.5, 00:00:10, Vlan58
     31.0.0.0/16 is subnetted, 4 subnets
R       31.3.0.0 [120/3] via 155.1.58.5, 00:00:10, Vlan58
R       31.2.0.0 [120/3] via 155.1.58.5, 00:00:10, Vlan58
R       31.1.0.0 [120/3] via 155.1.58.5, 00:00:10, Vlan58
R       31.0.0.0 [120/3] via 155.1.58.5, 00:00:10, Vlan58
     150.1.0.0/24 is subnetted, 10 subnets
R       150.1.7.0 [120/3] via 155.1.58.5, 00:00:10, Vlan58
R       150.1.6.0 [120/3] via 155.1.58.5, 00:00:11, Vlan58
R       150.1.5.0 [120/1] via 155.1.58.5, 00:00:11, Vlan58
R       150.1.4.0 [120/2] via 155.1.58.5, 00:00:11, Vlan58
R       150.1.3.0 [120/2] via 155.1.58.5, 00:00:11, Vlan58
R       150.1.2.0 [120/2] via 155.1.58.5, 00:00:11, Vlan58
R       150.1.1.0 [120/2] via 155.1.58.5, 00:00:11, Vlan58
R       150.1.10.0 [120/1] via 155.1.108.10, 00:00:11, Port-channel1
R       150.1.9.0 [120/4] via 155.1.58.5, 00:00:11, Vlan58
     30.0.0.0/16 is subnetted, 4 subnets
R       30.2.0.0 [120/3] via 155.1.58.5, 00:00:11, Vlan58
R       30.3.0.0 [120/3] via 155.1.58.5, 00:00:11, Vlan58
R       30.0.0.0 [120/3] via 155.1.58.5, 00:00:11, Vlan58
R       30.1.0.0 [120/3] via 155.1.58.5, 00:00:11, Vlan58
```

R2 will only have reachability information about connected networks of R5 and the routes learned from R5's link to SW2, since routes from R1, R3, and R4 cannot come in the Frame Relay of R5 and go back out the Frame Relay to reach R2.

```
Rack1R2#show ip route rip
     155.1.0.0/24 is subnetted, 7 subnets
R       155.1.10.0 [120/3] via 155.1.0.5, 00:00:07, Serial0/0.1
R       155.1.8.0 [120/2] via 155.1.0.5, 00:00:07, Serial0/0.1
R       155.1.5.0 [120/1] via 155.1.0.5, 00:00:07, Serial0/0.1
R       155.1.58.0 [120/1] via 155.1.0.5, 00:00:07, Serial0/0.1
R       155.1.45.0 [120/1] via 155.1.0.5, 00:00:07, Serial0/0.1
R       155.1.108.0 [120/2] via 155.1.0.5, 00:00:07, Serial0/0.1
     150.1.0.0/24 is subnetted, 4 subnets
R       150.1.5.0 [120/1] via 155.1.0.5, 00:00:07, Serial0/0.1
R       150.1.10.0 [120/3] via 155.1.0.5, 00:00:07, Serial0/0.1
R       150.1.8.0 [120/2] via 155.1.0.5, 00:00:07, Serial0/0.1
```

R1, R3, R6, SW1, and SW3 will not have reachability information about the link to BB2 or the routes learned from BB3, since R1 and R3 cannot learn the routes R2 and R4 send to R5 over the Frame Relay network.

```
Rack1SW3#show ip route rip
     155.1.0.0/24 is subnetted, 14 subnets
R       155.1.146.0 [120/2] via 155.1.79.7, 00:00:04, Vlan79
R       155.1.10.0 [120/5] via 155.1.79.7, 00:00:04, Vlan79
R       155.1.8.0 [120/4] via 155.1.79.7, 00:00:04, Vlan79
R       155.1.13.0 [120/2] via 155.1.79.7, 00:00:04, Vlan79
R       155.1.0.0 [120/2] via 155.1.79.7, 00:00:04, Vlan79
R       155.1.7.0 [120/1] via 155.1.79.7, 00:00:04, Vlan79
R       155.1.5.0 [120/3] via 155.1.79.7, 00:00:04, Vlan79
R       155.1.58.0 [120/3] via 155.1.79.7, 00:00:04, Vlan79
R       155.1.45.0 [120/3] via 155.1.79.7, 00:00:04, Vlan79
R       155.1.37.0 [120/1] via 155.1.79.7, 00:00:04, Vlan79
R       155.1.67.0 [120/1] via 155.1.79.7, 00:00:04, Vlan79
R       155.1.108.0 [120/4] via 155.1.79.7, 00:00:04, Vlan79
     54.0.0.0/24 is subnetted, 1 subnets
R       54.1.1.0 [120/2] via 155.1.79.7, 00:00:04, Vlan79
R    212.18.1.0/24 [120/3] via 155.1.79.7, 00:00:04, Vlan79
R    212.18.0.0/24 [120/3] via 155.1.79.7, 00:00:04, Vlan79
R    212.18.3.0/24 [120/3] via 155.1.79.7, 00:00:04, Vlan79
R    212.18.2.0/24 [120/3] via 155.1.79.7, 00:00:05, Vlan79
     150.1.0.0/24 is subnetted, 8 subnets
R       150.1.7.0 [120/1] via 155.1.79.7, 00:00:05, Vlan79
R       150.1.6.0 [120/2] via 155.1.79.7, 00:00:05, Vlan79
R       150.1.5.0 [120/3] via 155.1.79.7, 00:00:05, Vlan79
R       150.1.3.0 [120/2] via 155.1.79.7, 00:00:05, Vlan79
R       150.1.1.0 [120/3] via 155.1.79.7, 00:00:05, Vlan79
R       150.1.10.0 [120/5] via 155.1.79.7, 00:00:05, Vlan79
R       150.1.8.0 [120/4] via 155.1.79.7, 00:00:05, Vlan79
```

4.2 RIPv2 Authentication

- Configure RIPv2 authentication on the Ethernet link between R2 and BB2.
- Use the MD5 key number 1 with a password of CISCO.
- R2 should be learning RIP routes from BB2.
- Configure clear-text RIP authentication on the segment between R1 and R6 using the password CCIE.

Configuration

```
R1:
key chain RIP
 key 1
  key-string CCIE
!
interface FastEthernet0/0
 ip rip authentication mode text
 ip rip authentication key-chain RIP

R2:
key chain RIP
 key 1
  key-string CISCO
!
interface FastEthernet0/0
 ip address 192.10.1.2 255.255.255.0
 ip rip authentication mode md5
 ip rip authentication key-chain RIP

R6:
key chain RIP
 key 1
  key-string CCIE
!
interface FastEthernet0/0.146
 ip rip authentication mode text
 ip rip authentication key-chain RIP
```

Verification

Note

Without authentication configured R2 receives updates from BB2 but cannot install them in the RIP database.

```
Rack1R2#debug ip rip
RIP protocol debugging is on
RIP: received packet with MD5 authentication
RIP: ignored v2 packet from 192.10.1.254 (invalid authentication)
```

Once authentication is enabled updates from BB2 are installed.

```
Rack1R2#config t
Enter configuration commands, one per line.  End with CNTL/Z.
Rack1R2(config)#key chain RIP
Rack1R2(config-keychain)#key 1
Rack1R2(config-keychain-key)#key-string CISCO
Rack1R2(config)#interface Fa0/0
Rack1R2(config-if)#ip rip authentication key-chain RIP
Rack1R2(config-if)#ip rip authentication mode md5
Rack1R2(config-if)#end
Rack1R2#
Rack1R2#show ip route rip
R    222.22.2.0/24 [120/7] via 192.10.1.254, 00:00:07, FastEthernet0/0
     155.1.0.0/24 is subnetted, 7 subnets
R       155.1.10.0 [120/3] via 155.1.0.5, 00:00:28, Serial0/0.1
R       155.1.8.0 [120/2] via 155.1.0.5, 00:00:28, Serial0/0.1
R       155.1.5.0 [120/1] via 155.1.0.5, 00:00:28, Serial0/0.1
R       155.1.58.0 [120/1] via 155.1.0.5, 00:00:28, Serial0/0.1
R       155.1.45.0 [120/1] via 155.1.0.5, 00:00:28, Serial0/0.1
R       155.1.108.0 [120/2] via 155.1.0.5, 00:00:28, Serial0/0.1
R    220.20.3.0/24 [120/7] via 192.10.1.254, 00:00:07, FastEthernet0/0
     150.1.0.0/24 is subnetted, 4 subnets
R       150.1.5.0 [120/1] via 155.1.0.5, 00:00:28, Serial0/0.1
R       150.1.10.0 [120/3] via 155.1.0.5, 00:00:28, Serial0/0.1
R       150.1.8.0 [120/2] via 155.1.0.5, 00:00:28, Serial0/0.1
R    205.90.31.0/24 [120/7] via 192.10.1.254, 00:00:07, FastEthernet0/0
```

☠ Pitfall

Whitespace counts as a valid character for key chain authentication. Use the **show key-chain** command to ensure that a space is not at the end of the authentication key for either RIP or EIGRP.

```
Rack1R2#conf t
Enter configuration commands, one per line.  End with CNTL/Z.
Rack1R2(config)#key chain RIP
Rack1R2(config-keychain)#key 1
Rack1R2(config-keychain-key)#key-string CISCO ?
LINE    <cr>

Rack1R2(config-keychain-key)#key-string CISCO
Rack1R2(config-keychain-key)#end
Rack1R2#

Rack1R2#show key chain
Key-chain RIP:
    key 1 -- text "CISCO "
        accept lifetime (always valid) - (always valid) [valid now]
        send lifetime (always valid) - (always valid) [valid now]

Rack1R2#debug ip rip
RIP protocol debugging is on
Rack1R2#
RIP: received packet with MD5 authentication
RIP: ignored v2 packet from 192.10.1.254 (invalid authentication)
```

4.3 RIPv2 Split Horizon

- Disable split-horizon on R5’s connection to the Frame Relay cloud.
- Test reachability to all networks and note any changes within the topology.

Configuration

```
R5:
interface Serial0/0.1 multipoint
 no ip split-horizon
```

Verification

> ✎ **Note**
>
> Now that R5 has split horizon disabled on the Frame Relay subinterface, R1, R2, and R3 can be advertised all updates, and all devices in the topology have full reachability information.

```
Rack1R2#show ip route rip
R    222.22.2.0/24 [120/7] via 192.10.1.254, 00:00:21, FastEthernet0/0
R    204.12.1.0/24 [120/2] via 155.1.0.4, 00:00:17, Serial0/0.1
     155.1.0.0/24 is subnetted, 14 subnets
R       155.1.146.0 [120/2] via 155.1.0.1, 00:00:17, Serial0/0.1
R       155.1.10.0 [120/3] via 155.1.0.5, 00:00:18, Serial0/0.1
R       155.1.8.0 [120/2] via 155.1.0.5, 00:00:18, Serial0/0.1
R       155.1.9.0 [120/4] via 155.1.0.3, 00:00:18, Serial0/0.1
R       155.1.13.0 [120/2] via 155.1.0.1, 00:00:17, Serial0/0.1
R       155.1.7.0 [120/3] via 155.1.0.3, 00:00:18, Serial0/0.1
R       155.1.5.0 [120/1] via 155.1.0.5, 00:00:18, Serial0/0.1
R       155.1.58.0 [120/1] via 155.1.0.5, 00:00:17, Serial0/0.1
R       155.1.45.0 [120/1] via 155.1.0.5, 00:00:17, Serial0/0.1
R       155.1.37.0 [120/2] via 155.1.0.3, 00:00:17, Serial0/0.1
R       155.1.79.0 [120/3] via 155.1.0.3, 00:00:17, Serial0/0.1
R       155.1.67.0 [120/3] via 155.1.0.1, 00:00:17, Serial0/0.1
R       155.1.108.0 [120/2] via 155.1.0.5, 00:00:17, Serial0/0.1
R    220.20.3.0/24 [120/7] via 192.10.1.254, 00:00:22, FastEthernet0/0
     54.0.0.0/24 is subnetted, 1 subnets
R       54.1.1.0 [120/3] via 155.1.0.1, 00:00:19, Serial0/0.1
R    212.18.1.0/24 [120/4] via 155.1.0.1, 00:00:19, Serial0/0.1
R    212.18.0.0/24 [120/4] via 155.1.0.1, 00:00:19, Serial0/0.1
R    212.18.3.0/24 [120/4] via 155.1.0.1, 00:00:19, Serial0/0.1
R    212.18.2.0/24 [120/4] via 155.1.0.1, 00:00:19, Serial0/0.1
     31.0.0.0/16 is subnetted, 4 subnets
R       31.3.0.0 [120/3] via 155.1.0.4, 00:00:19, Serial0/0.1
R       31.2.0.0 [120/3] via 155.1.0.4, 00:00:19, Serial0/0.1
R       31.1.0.0 [120/3] via 155.1.0.4, 00:00:19, Serial0/0.1
R       31.0.0.0 [120/3] via 155.1.0.4, 00:00:19, Serial0/0.1
     150.1.0.0/24 is subnetted, 10 subnets
R       150.1.7.0 [120/3] via 155.1.0.3, 00:00:19, Serial0/0.1
R       150.1.6.0 [120/3] via 155.1.0.1, 00:00:19, Serial0/0.1
R       150.1.5.0 [120/1] via 155.1.0.5, 00:00:19, Serial0/0.1
R       150.1.4.0 [120/2] via 155.1.0.4, 00:00:20, Serial0/0.1
R       150.1.3.0 [120/2] via 155.1.0.3, 00:00:20, Serial0/0.1
R       150.1.1.0 [120/2] via 155.1.0.1, 00:00:20, Serial0/0.1
R       150.1.10.0 [120/3] via 155.1.0.5, 00:00:20, Serial0/0.1
R       150.1.9.0 [120/4] via 155.1.0.3, 00:00:20, Serial0/0.1
R       150.1.8.0 [120/2] via 155.1.0.5, 00:00:20, Serial0/0.1
R    205.90.31.0/24 [120/7] via 192.10.1.254, 00:00:23, FastEthernet0/0
     30.0.0.0/16 is subnetted, 4 subnets
R       30.2.0.0 [120/3] via 155.1.0.4, 00:00:20, Serial0/0.1
R       30.3.0.0 [120/3] via 155.1.0.4, 00:00:20, Serial0/0.1
R       30.0.0.0 [120/3] via 155.1.0.4, 00:00:20, Serial0/0.1
R       30.1.0.0 [120/3] via 155.1.0.4, 00:00:20, Serial0/0.1
```

4.4 RIPv2 Auto-Summary

- Enable auto-summary under the RIP process of R4.
- Note any changes in the network advertisements that R4 is sending.

Configuration

```
R4:
router rip
 auto-summary
```

Verification

> **Note**
>
> Before auto-summary is enabled:

```
Rack1R4#debug ip rip
RIP protocol debugging is on
RIP: sending v2 update to 224.0.0.9 via Serial0/1 (155.1.45.4)
RIP: build update entries
        30.0.0.0/16 via 0.0.0.0, metric 2, tag 0
        30.1.0.0/16 via 0.0.0.0, metric 2, tag 0
        30.2.0.0/16 via 0.0.0.0, metric 2, tag 0
        30.3.0.0/16 via 0.0.0.0, metric 2, tag 0
        31.0.0.0/16 via 0.0.0.0, metric 2, tag 0
        31.1.0.0/16 via 0.0.0.0, metric 2, tag 0
        31.2.0.0/16 via 0.0.0.0, metric 2, tag 0
        31.3.0.0/16 via 0.0.0.0, metric 2, tag 0
        150.1.4.0/24 via 0.0.0.0, metric 1, tag 0
        155.1.0.0/24 via 0.0.0.0, metric 1, tag 0
        204.12.1.0/24 via 0.0.0.0, metric 1, tag 0
RIP: sending v2 update to 224.0.0.9 via Serial0/0.1 (155.1.0.4)
RIP: build update entries
        30.0.0.0/16 via 0.0.0.0, metric 2, tag 0
        30.1.0.0/16 via 0.0.0.0, metric 2, tag 0
        30.2.0.0/16 via 0.0.0.0, metric 2, tag 0
        30.3.0.0/16 via 0.0.0.0, metric 2, tag 0
        31.0.0.0/16 via 0.0.0.0, metric 2, tag 0
        31.1.0.0/16 via 0.0.0.0, metric 2, tag 0
        31.2.0.0/16 via 0.0.0.0, metric 2, tag 0
        31.3.0.0/16 via 0.0.0.0, metric 2, tag 0
        150.1.4.0/24 via 0.0.0.0, metric 1, tag 0
        155.1.45.0/24 via 0.0.0.0, metric 1, tag 0
        204.12.1.0/24 via 0.0.0.0, metric 1, tag 0
```

After auto-summary is enabled R4 summarizes the routes learned from BB3 and its own Loopback0 network out to R5, and summarizes the 155.1.0.0/16 transit network, its own Loopback0 network, and the link to BB1 out to BB3.

```
RIP: sending v2 update to 224.0.0.9 via Serial0/1 (155.1.45.4)
RIP: build update entries
        30.0.0.0/8 via 0.0.0.0, metric 2, tag 0
        31.0.0.0/8 via 0.0.0.0, metric 2, tag 0
        150.1.0.0/16 via 0.0.0.0, metric 1, tag 0
        155.1.0.0/24 via 0.0.0.0, metric 1, tag 0
        204.12.1.0/24 via 0.0.0.0, metric 1, tag 0
RIP: sending v2 update to 224.0.0.9 via Serial0/0.1 (155.1.0.4)
RIP: build update entries
        30.0.0.0/8 via 0.0.0.0, metric 2, tag 0
        31.0.0.0/8 via 0.0.0.0, metric 2, tag 0
        150.1.0.0/16 via 0.0.0.0, metric 1, tag 0
        155.1.45.0/24 via 0.0.0.0, metric 1, tag 0
        204.12.1.0/24 via 0.0.0.0, metric 1, tag 0
RIP: sending v2 update to 224.0.0.9 via FastEthernet0/0 (204.12.1.4)
RIP: build update entries
        54.0.0.0/8 via 0.0.0.0, metric 4, tag 0
        150.1.0.0/16 via 0.0.0.0, metric 1, tag 0
        155.1.0.0/16 via 0.0.0.0, metric 1, tag 0
        192.10.1.0/24 via 0.0.0.0, metric 3, tag 0
        205.90.31.0/24 via 0.0.0.0, metric 10, tag 0
        212.18.0.0/24 via 0.0.0.0, metric 5, tag 0
        212.18.1.0/24 via 0.0.0.0, metric 5, tag 0
        212.18.2.0/24 via 0.0.0.0, metric 5, tag 0
        212.18.3.0/24 via 0.0.0.0, metric 5, tag 0
        220.20.3.0/24 via 0.0.0.0, metric 10, tag 0
        222.22.2.0/24 via 0.0.0.0, metric 10, tag 0
```

4.5 RIP Send and Receive Versions

- Remove the **`version 2`** commands under the RIP processes of SW2 and SW4.
- Configure SW2 to send and receive only RIPv2 updates on VLAN 58.
- Configure SW2 to send and receive only RIPv1 updates on the link to SW4.
- Note any changes in reachability or routing information in the network.

Configuration

```
SW2:
interface Vlan58
 ip rip send version 2
 ip rip receive version 2
!
interface Port-channel1
 ip rip send version 1
 ip rip receive version 1
!
router rip
 network 150.1.0.0
 network 155.1.0.0
 no auto-summary

SW4:
router rip
 network 150.1.0.0
 network 155.1.0.0
 no auto-summary
```

Verification

> **Note**
>
> SW2 and SW4's routing tables before the change to RIPv1.

```
Rack1SW2#show ip route rip
R    222.22.2.0/24 [120/9] via 155.1.58.5, 00:00:15, Vlan58
R    204.12.1.0/24 [120/2] via 155.1.58.5, 00:00:15, Vlan58
     155.1.0.0/24 is subnetted, 14 subnets
R       155.1.146.0 [120/2] via 155.1.58.5, 00:00:15, Vlan58
R       155.1.10.0 [120/1] via 155.1.108.10, 00:00:24, Port-channel1
R       155.1.9.0 [120/4] via 155.1.58.5, 00:00:15, Vlan58
R       155.1.13.0 [120/2] via 155.1.58.5, 00:00:15, Vlan58
R       155.1.0.0 [120/1] via 155.1.58.5, 00:00:15, Vlan58
R       155.1.7.0 [120/3] via 155.1.58.5, 00:00:15, Vlan58
R       155.1.5.0 [120/1] via 155.1.58.5, 00:00:15, Vlan58
R       155.1.45.0 [120/1] via 155.1.58.5, 00:00:15, Vlan58
R       155.1.37.0 [120/2] via 155.1.58.5, 00:00:15, Vlan58
R       155.1.79.0 [120/3] via 155.1.58.5, 00:00:15, Vlan58
R       155.1.67.0 [120/3] via 155.1.58.5, 00:00:15, Vlan58
R    220.20.3.0/24 [120/9] via 155.1.58.5, 00:00:15, Vlan58
     54.0.0.0/24 is subnetted, 1 subnets
R       54.1.1.0 [120/3] via 155.1.58.5, 00:00:15, Vlan58
R    212.18.1.0/24 [120/4] via 155.1.58.5, 00:00:15, Vlan58
R    212.18.0.0/24 [120/4] via 155.1.58.5, 00:00:16, Vlan58
R    212.18.3.0/24 [120/4] via 155.1.58.5, 00:00:16, Vlan58
R    212.18.2.0/24 [120/4] via 155.1.58.5, 00:00:16, Vlan58
R    192.10.1.0/24 [120/2] via 155.1.58.5, 00:00:16, Vlan58
     31.0.0.0/8 is subnetted, 1 subnets
R       31.0.0.0 [120/3] via 155.1.58.5, 00:00:16, Vlan58
     150.1.0.0/16 is variably subnetted, 10 subnets, 2 masks
R       150.1.7.0/24 [120/3] via 155.1.58.5, 00:00:16, Vlan58
R       150.1.6.0/24 [120/3] via 155.1.58.5, 00:00:16, Vlan58
R       150.1.5.0/24 [120/1] via 155.1.58.5, 00:00:16, Vlan58
R       150.1.3.0/24 [120/2] via 155.1.58.5, 00:00:16, Vlan58
R       150.1.2.0/24 [120/2] via 155.1.58.5, 00:00:16, Vlan58
R       150.1.1.0/24 [120/2] via 155.1.58.5, 00:00:16, Vlan58
R       150.1.0.0/16 [120/2] via 155.1.58.5, 00:00:16, Vlan58
R       150.1.10.0/24 [120/1] via 155.1.108.10, 00:00:26, Port-channel1
R       150.1.9.0/24 [120/4] via 155.1.58.5, 00:00:16, Vlan58
R    205.90.31.0/24 [120/9] via 155.1.58.5, 00:00:17, Vlan58
     30.0.0.0/8 is subnetted, 1 subnets
R       30.0.0.0 [120/3] via 155.1.58.5, 00:00:17, Vlan58
```

```
Rack1SW4#show ip route rip
R    222.22.2.0/24 [120/10] via 155.1.108.8, 00:00:07, Port-channel1
R    204.12.1.0/24 [120/3] via 155.1.108.8, 00:00:07, Port-channel1
     155.1.0.0/24 is subnetted, 14 subnets
R       155.1.146.0 [120/3] via 155.1.108.8, 00:00:07, Port-channel1
R       155.1.8.0 [120/1] via 155.1.108.8, 00:00:07, Port-channel1
R       155.1.9.0 [120/5] via 155.1.108.8, 00:00:07, Port-channel1
R       155.1.13.0 [120/3] via 155.1.108.8, 00:00:07, Port-channel1
R       155.1.0.0 [120/2] via 155.1.108.8, 00:00:08, Port-channel1
R       155.1.7.0 [120/4] via 155.1.108.8, 00:00:08, Port-channel1
R       155.1.5.0 [120/2] via 155.1.108.8, 00:00:08, Port-channel1
R       155.1.58.0 [120/1] via 155.1.108.8, 00:00:08, Port-channel1
R       155.1.45.0 [120/2] via 155.1.108.8, 00:00:08, Port-channel1
R       155.1.37.0 [120/3] via 155.1.108.8, 00:00:08, Port-channel1
R       155.1.79.0 [120/4] via 155.1.108.8, 00:00:08, Port-channel1
R       155.1.67.0 [120/4] via 155.1.108.8, 00:00:08, Port-channel1
R    220.20.3.0/24 [120/10] via 155.1.108.8, 00:00:08, Port-channel1
     54.0.0.0/24 is subnetted, 1 subnets
R       54.1.1.0 [120/4] via 155.1.108.8, 00:00:08, Port-channel1
R    212.18.1.0/24 [120/5] via 155.1.108.8, 00:00:08, Port-channel1
R    212.18.0.0/24 [120/5] via 155.1.108.8, 00:00:08, Port-channel1
R    212.18.3.0/24 [120/5] via 155.1.108.8, 00:00:08, Port-channel1
R    212.18.2.0/24 [120/5] via 155.1.108.8, 00:00:08, Port-channel1
R    192.10.1.0/24 [120/3] via 155.1.108.8, 00:00:09, Port-channel1
     31.0.0.0/8 is subnetted, 1 subnets
R       31.0.0.0 [120/4] via 155.1.108.8, 00:00:09, Port-channel1
     150.1.0.0/16 is variably subnetted, 10 subnets, 2 masks
R       150.1.7.0/24 [120/4] via 155.1.108.8, 00:00:09, Port-channel1
R       150.1.6.0/24 [120/4] via 155.1.108.8, 00:00:09, Port-channel1
R       150.1.5.0/24 [120/2] via 155.1.108.8, 00:00:09, Port-channel1
R       150.1.3.0/24 [120/3] via 155.1.108.8, 00:00:09, Port-channel1
R       150.1.2.0/24 [120/3] via 155.1.108.8, 00:00:09, Port-channel1
R       150.1.1.0/24 [120/3] via 155.1.108.8, 00:00:09, Port-channel1
R       150.1.0.0/16 [120/3] via 155.1.108.8, 00:00:09, Port-channel1
R       150.1.9.0/24 [120/5] via 155.1.108.8, 00:00:09, Port-channel1
R       150.1.8.0/24 [120/1] via 155.1.108.8, 00:00:09, Port-channel1
R    205.90.31.0/24 [120/10] via 155.1.108.8, 00:00:09, Port-channel1
     30.0.0.0/8 is subnetted, 1 subnets
R       30.0.0.0 [120/4] via 155.1.108.8, 00:00:09, Port-channel1
```

After RIPv1 between SW2 and SW4.

```
Rack1SW2#show ip route rip
R    222.22.2.0/24 [120/9] via 155.1.58.5, 00:00:18, Vlan58
R    204.12.1.0/24 [120/2] via 155.1.58.5, 00:00:18, Vlan58
     155.1.0.0/24 is subnetted, 14 subnets
R       155.1.146.0 [120/2] via 155.1.58.5, 00:00:18, Vlan58
R       155.1.10.0 [120/1] via 155.1.108.10, 00:00:03, Port-channel1
R       155.1.9.0 [120/4] via 155.1.58.5, 00:00:18, Vlan58
R       155.1.13.0 [120/2] via 155.1.58.5, 00:00:18, Vlan58
R       155.1.0.0 [120/1] via 155.1.58.5, 00:00:18, Vlan58
R       155.1.7.0 [120/3] via 155.1.58.5, 00:00:18, Vlan58
R       155.1.5.0 [120/1] via 155.1.58.5, 00:00:19, Vlan58
R       155.1.45.0 [120/1] via 155.1.58.5, 00:00:19, Vlan58
R       155.1.37.0 [120/2] via 155.1.58.5, 00:00:19, Vlan58
R       155.1.79.0 [120/3] via 155.1.58.5, 00:00:19, Vlan58
R       155.1.67.0 [120/3] via 155.1.58.5, 00:00:19, Vlan58
R    220.20.3.0/24 [120/9] via 155.1.58.5, 00:00:19, Vlan58
     54.0.0.0/24 is subnetted, 1 subnets
R       54.1.1.0 [120/3] via 155.1.58.5, 00:00:19, Vlan58
R    212.18.1.0/24 [120/4] via 155.1.58.5, 00:00:19, Vlan58
R    212.18.0.0/24 [120/4] via 155.1.58.5, 00:00:19, Vlan58
R    212.18.3.0/24 [120/4] via 155.1.58.5, 00:00:19, Vlan58
R    212.18.2.0/24 [120/4] via 155.1.58.5, 00:00:19, Vlan58
R    192.10.1.0/24 [120/2] via 155.1.58.5, 00:00:19, Vlan58
R    31.0.0.0/8 [120/3] via 155.1.58.5, 00:00:19, Vlan58
     150.1.0.0/16 is variably subnetted, 9 subnets, 2 masks
R       150.1.7.0/24 [120/3] via 155.1.58.5, 00:00:19, Vlan58
R       150.1.6.0/24 [120/3] via 155.1.58.5, 00:00:19, Vlan58
R       150.1.5.0/24 [120/1] via 155.1.58.5, 00:00:20, Vlan58
R       150.1.3.0/24 [120/2] via 155.1.58.5, 00:00:20, Vlan58
R       150.1.2.0/24 [120/2] via 155.1.58.5, 00:00:20, Vlan58
R       150.1.1.0/24 [120/2] via 155.1.58.5, 00:00:20, Vlan58
R       150.1.0.0/16 [120/2] via 155.1.58.5, 00:00:20, Vlan58
R       150.1.9.0/24 [120/4] via 155.1.58.5, 00:00:20, Vlan58
R    205.90.31.0/24 [120/9] via 155.1.58.5, 00:00:20, Vlan58
R    30.0.0.0/8 [120/3] via 155.1.58.5, 00:00:20, Vlan58
```

```
Rack1SW4#show ip route rip
R    222.22.2.0/24 [120/10] via 155.1.108.8, 00:00:17, Port-channel1
R    204.12.1.0/24 [120/3] via 155.1.108.8, 00:00:17, Port-channel1
     155.1.0.0/24 is subnetted, 14 subnets
R       155.1.146.0 [120/3] via 155.1.108.8, 00:00:17, Port-channel1
R       155.1.8.0 [120/1] via 155.1.108.8, 00:00:17, Port-channel1
R       155.1.9.0 [120/5] via 155.1.108.8, 00:00:17, Port-channel1
R       155.1.13.0 [120/3] via 155.1.108.8, 00:00:17, Port-channel1
R       155.1.0.0 [120/2] via 155.1.108.8, 00:00:17, Port-channel1
R       155.1.7.0 [120/4] via 155.1.108.8, 00:00:17, Port-channel1
R       155.1.5.0 [120/2] via 155.1.108.8, 00:00:17, Port-channel1
R       155.1.58.0 [120/1] via 155.1.108.8, 00:00:17, Port-channel1
R       155.1.45.0 [120/2] via 155.1.108.8, 00:00:17, Port-channel1
R       155.1.37.0 [120/3] via 155.1.108.8, 00:00:17, Port-channel1
R       155.1.79.0 [120/4] via 155.1.108.8, 00:00:17, Port-channel1
R       155.1.67.0 [120/4] via 155.1.108.8, 00:00:17, Port-channel1
R    220.20.3.0/24 [120/10] via 155.1.108.8, 00:00:18, Port-channel1
R    54.0.0.0/8 [120/4] via 155.1.108.8, 00:00:18, Port-channel1
R    212.18.1.0/24 [120/5] via 155.1.108.8, 00:00:18, Port-channel1
R    212.18.0.0/24 [120/5] via 155.1.108.8, 00:00:18, Port-channel1
R    212.18.3.0/24 [120/5] via 155.1.108.8, 00:00:18, Port-channel1
R    212.18.2.0/24 [120/5] via 155.1.108.8, 00:00:18, Port-channel1
R    192.10.1.0/24 [120/3] via 155.1.108.8, 00:00:18, Port-channel1
R    31.0.0.0/8 [120/4] via 155.1.108.8, 00:00:18, Port-channel1
R    205.90.31.0/24 [120/10] via 155.1.108.8, 00:00:18, Port-channel1
R    30.0.0.0/8 [120/4] via 155.1.108.8, 00:00:18, Port-channel1
```

Since SW2 and SW4 are running only RIPv1 between them, which does not include subnet mask information in the update, only classful networks with their classful masks can be advertised, along with contiguous subnets that share the same major network and subnet mask as the transit link between the neighbors. This is the reason why SW4 still learns about all the 155.1.X.0/24 networks, but cannot learn any of the 150.1.X.0/24 Loopback networks. Likewise SW2 cannot learn the 150.1.10.0/24 network from SW4. When traffic is sent to 155.1.10.10 it is routed based on the RIPv2 summary generated by R4. Since R4 does not have a longer match to the 150.1.10.10 destination, traffic is dropped and an ICMP unreachable message is sent back to the originating host. This problem is inherent to the design of RIPv1 and cannot be fixed in this topology without renumbering subnets or running RIPv2.

```
Rack1SW4#show ip route 150.1.0.0
Routing entry for 150.1.0.0/24, 1 known subnets
  Attached (1 connections)
  Redistributing via rip

C       150.1.10.0 is directly connected, Loopback0

Rack1SW2#debug ip icmp
ICMP packet debugging is on
Rack1SW2#ping 150.1.10.10

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 150.1.10.10, timeout is 2 seconds:
U.U.U
Success rate is 0 percent (0/5)
ICMP: dst (155.1.58.8) host unreachable rcv from 155.1.0.4
ICMP: dst (155.1.58.8) host unreachable rcv from 155.1.0.4
ICMP: dst (155.1.58.8) host unreachable rcv from 155.1.0.4

Rack1SW4#debug ip packet
IP packet debugging is on
Rack1SW4#ping 150.1.8.8

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 150.1.8.8, timeout is 2 seconds:
.....
Success rate is 0 percent (0/5)
IP: s=150.1.10.10 (local), d=150.1.8.8, len 100, unroutable.
IP: s=150.1.10.10 (local), d=150.1.8.8, len 100, unroutable.
IP: s=150.1.10.10 (local), d=150.1.8.8, len 100, unroutable
IP: s=150.1.10.10 (local), d=150.1.8.8, len 100, unroutable
IP: s=150.1.10.10 (local), d=150.1.8.8, len 100, unroutable
```

While RIPv1 will not be tested on in the CCIE Lab Exam, understanding the problems with legacy protocol design can assist you in understanding the “why” behind routing logic for IPv4.

4.6 RIPv2 Manual Summarization

- Configure R4 to send two summary routes for the RIP networks learned from BB3 to R5.
- Ensure that these summaries do not overlap any address space that R4 does not have a longer match route to.

Configuration

```
R4:
interface Serial0/0.1 point-to-point
 ip summary-address rip 30.0.0.0 255.252.0.0
 ip summary-address rip 31.0.0.0 255.252.0.0
!
interface Serial0/1
 ip summary-address rip 30.0.0.0 255.252.0.0
 ip summary-address rip 31.0.0.0 255.252.0.0
!
router rip
 no auto-summary
```

Verification

```
Rack1R5#show ip route 30.0.0.0
Routing entry for 30.0.0.0/14, 1 known subnets
  Redistributing via rip

R       30.0.0.0 [120/2] via 155.1.45.4, 00:00:04, Serial0/1
                 [120/2] via 155.1.0.4, 00:00:00, Serial0/0.1

Rack1R5#show ip route 31.0.0.0
Routing entry for 31.0.0.0/14, 1 known subnets
  Redistributing via rip

R       31.0.0.0 [120/2] via 155.1.45.4, 00:00:11, Serial0/1
                 [120/2] via 155.1.0.4, 00:00:06, Serial0/0.1
```

4.7 RIPv2 Convergence Timers

- Change the RIP timers throughout the topology to make convergence three times faster than the default.
- Ensure that this configuration does not affect the links to the BB routers.

Configuration

```
R1:
router rip
 version 2
 timers basic 10 60 60 80

R2:
interface FastEthernet0/0
 ip rip advertise 30
!
router rip
 version 2
 timers basic 10 60 60 80

R3:
router rip
 version 2
 timers basic 10 60 60 80

R4:
interface FastEthernet0/0
 ip rip advertise 30
!
router rip
 version 2
 timers basic 10 60 60 80

R5:
router rip
 version 2
 timers basic 10 60 60 80

R6:
interface Serial0/0
 ip rip advertise 30
!
router rip
 version 2
 timers basic 10 60 60 80

SW1:
router rip
 version 2
 timers basic 10 60 60 80
```

```
SW2:
router rip
 version 2
 timers basic 10 60 60 80

SW3:
router rip
 version 2
 timers basic 10 60 60 80

SW4:
router rip
 version 2
 timers basic 10 60 60 80
```

Verification

Note

Prior to changing RIP timers.

```
Rack1R1#show ip protocols | include seconds
  Sending updates every 30 seconds, next due in 22 seconds
  Invalid after 180 seconds, hold down 180, flushed after 240
```

After changing RIP timers:

```
Rack1R1#show ip protocols | include seconds
  Sending updates every 10 seconds, next due in 9 seconds
  Invalid after 60 seconds, hold down 60, flushed after 80
```

4.8 RIPv2 Offset List

- Configure an offset-list on R6 so that all traffic going to BB1 uses the Ethernet link to SW1.
- If this link is down traffic should be rerouted over the link to R1.

Configuration

```
R6:
router rip
 offset-list 0 out 5 FastEthernet0/0.146
```

Verification

Note

Before metric offset.

```
Rack1R2#traceroute 212.18.0.1

Type escape sequence to abort.
Tracing the route to 212.18.0.1

  1 155.1.0.5 28 msec 28 msec 28 msec
  2 155.1.0.1 64 msec 56 msec 56 msec
  3 155.1.146.6 56 msec 57 msec 56 msec
  4 54.1.1.254 76 msec *  76 msec
```

After metric offset,

```
Rack1R2#traceroute 212.18.0.1

Type escape sequence to abort.
Tracing the route to 212.18.0.1

  1 155.1.0.5 28 msec 32 msec 28 msec
  2 155.1.0.3 64 msec 56 msec 56 msec
  3 155.1.37.7 56 msec 56 msec 60 msec
  4 155.1.67.6 56 msec 57 msec 56 msec
  5 54.1.1.254 72 msec *  72 msec
```


Traffic should reroute to R1 if SW1 fails.

```
Rack1SW1#conf t
Enter configuration commands, one per line.  End with CNTL/Z.
Rack1SW1(config)#interface Vlan67
Rack1SW1(config-if)#shut
Rack1SW1(config-if)#end
Rack1SW1#

Rack1R2#traceroute 212.18.0.1

Type escape sequence to abort.
Tracing the route to 212.18.0.1

  1 155.1.0.5 28 msec 28 msec 28 msec
  2 155.1.0.1 68 msec 56 msec 56 msec
  3 155.1.146.6 56 msec 56 msec 56 msec
  4 54.1.1.254 76 msec *  72 msec
```

4.9 RIPv2 Filtering with Passive Interface

- Configure the passive interface feature on SW2 so that it learns RIP updates from SW4 but does not advertise any information back to SW4.

Configuration

```
SW2:
router rip
 passive-interface Port-channel1
```

Verification

```
Rack1SW2#show ip route | include Port-channel1
R       155.1.10.0 [120/1] via 155.1.108.10, 00:00:07, Port-channel1
C       155.1.108.0 is directly connected, Port-channel1

Rack1SW4#show ip route
Codes: C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default, U - per-user static route
       o - ODR, P - periodic downloaded static route

Gateway of last resort is not set

     155.1.0.0/24 is subnetted, 2 subnets
C       155.1.10.0 is directly connected, Vlan10
C       155.1.108.0 is directly connected, Port-channel1
     150.1.0.0/24 is subnetted, 1 subnets
C       150.1.10.0 is directly connected, Loopback0
```

4.10 RIPv2 Filtering with Prefix-Lists

- Configure a prefix-list on R5 so that it does not advertise the two RIP summaries of the BB3 networks to SW2, and so that all other networks are allowed to be advertised.
- Configure a prefix-list on R5 so that it does not install any updates received from R4 on the Frame Relay network. This configuration should not affect the updates received from other neighbors on this segment.
- Ensure that route feedback does not occur for the summaries that R4 is generating to R5.

Configuration

```
R4:
ip route 30.0.0.0 255.252.0.0 Null0
ip route 31.0.0.0 255.252.0.0 Null0

R5:
router rip
 distribute-list prefix RIP_FILTER_TO_SW2 out FastEthernet0/0
 distribute-list prefix PERMIT_ALL gateway NOT_FROM_R4 in
!
ip prefix-list NOT_FROM_R4 seq 5 deny 155.1.0.4/32
ip prefix-list NOT_FROM_R4 seq 10 permit 0.0.0.0/0 le 32
!
ip prefix-list PERMIT_ALL seq 5 permit 0.0.0.0/0 le 32
!
ip prefix-list RIP_FILTER_TO_SW2 seq 5 deny 30.0.0.0/14
ip prefix-list RIP_FILTER_TO_SW2 seq 10 deny 31.0.0.0/14
ip prefix-list RIP_FILTER_TO_SW2 seq 15 permit 0.0.0.0/0 le 32
```

Verification

```
Rack1R5#show ip route rip
R    222.22.2.0/24 [120/8] via 155.1.0.2, 00:00:05, Serial0/0.1
R    204.12.1.0/24 [120/1] via 155.1.45.4, 00:00:01, Serial0/1
     155.1.0.0/24 is subnetted, 14 subnets
R       155.1.146.0 [120/1] via 155.1.0.1, 00:00:02, Serial0/0.1
R       155.1.10.0 [120/2] via 155.1.58.8, 00:00:02, FastEthernet0/0
R       155.1.8.0 [120/1] via 155.1.58.8, 00:00:02, FastEthernet0/0
R       155.1.9.0 [120/3] via 155.1.0.3, 00:00:08, Serial0/0.1
R       155.1.13.0 [120/1] via 155.1.0.3, 00:00:08, Serial0/0.1
                   [120/1] via 155.1.0.1, 00:00:02, Serial0/0.1
R       155.1.7.0 [120/2] via 155.1.0.3, 00:00:08, Serial0/0.1
R       155.1.37.0 [120/1] via 155.1.0.3, 00:00:08, Serial0/0.1
R       155.1.79.0 [120/2] via 155.1.0.3, 00:00:08, Serial0/0.1
R       155.1.67.0 [120/2] via 155.1.0.3, 00:00:08, Serial0/0.1
R       155.1.108.0 [120/1] via 155.1.58.8, 00:00:02, FastEthernet0/0
R    220.20.3.0/24 [120/8] via 155.1.0.2, 00:00:05, Serial0/0.1
     54.0.0.0/24 is subnetted, 1 subnets
R       54.1.1.0 [120/3] via 155.1.0.3, 00:00:08, Serial0/0.1
R    212.18.1.0/24 [120/4] via 155.1.0.3, 00:00:08, Serial0/0.1
R    212.18.0.0/24 [120/4] via 155.1.0.3, 00:00:08, Serial0/0.1
```

```
R    212.18.3.0/24 [120/4] via 155.1.0.3, 00:00:08, Serial0/0.1
R    212.18.2.0/24 [120/4] via 155.1.0.3, 00:00:08, Serial0/0.1
R    192.10.1.0/24 [120/1] via 155.1.0.2, 00:00:05, Serial0/0.1
     31.0.0.0/14 is subnetted, 1 subnets
R       31.0.0.0 [120/2] via 155.1.45.4, 00:00:01, Serial0/1
     150.1.0.0/24 is subnetted, 9 subnets
R       150.1.7.0 [120/2] via 155.1.0.3, 00:00:08, Serial0/0.1
R       150.1.6.0 [120/3] via 155.1.0.3, 00:00:08, Serial0/0.1
R       150.1.4.0 [120/1] via 155.1.45.4, 00:00:01, Serial0/1
R       150.1.3.0 [120/1] via 155.1.0.3, 00:00:08, Serial0/0.1
R       150.1.2.0 [120/1] via 155.1.0.2, 00:00:05, Serial0/0.1
R       150.1.1.0 [120/1] via 155.1.0.1, 00:00:02, Serial0/0.1
R       150.1.9.0 [120/3] via 155.1.0.3, 00:00:08, Serial0/0.1
R       150.1.8.0 [120/1] via 155.1.58.8, 00:00:03, FastEthernet0/0
R    205.90.31.0/24 [120/8] via 155.1.0.2, 00:00:05, Serial0/0.1
     30.0.0.0/14 is subnetted, 1 subnets
R       30.0.0.0 [120/2] via 155.1.45.4, 00:00:01, Serial0/1

Rack1SW2#show ip route 30.0.0.0
% Network not in table

Rack1SW2#show ip route 31.0.0.0
% Network not in table

Rack1R2#show ip route 30.0.0.0
Routing entry for 30.0.0.0/14, 1 known subnets
  Redistributing via rip

R       30.0.0.0 [120/3] via 155.1.0.5, 00:00:04, Serial0/0.1

Rack1R2#show ip route 31.0.0.0
Routing entry for 31.0.0.0/14, 1 known subnets
  Redistributing via rip

R       31.0.0.0 [120/3] via 155.1.0.5, 00:00:08, Serial0/0.1
```

Note

The first filter on R5, out to SW2, denies the exact to prefixes that are the summaries coming from R4, and permits all other routes. The syntax **`0.0.0.0/0 le 32`** in a prefix-list means match all routes. The next filter is done based on both the routes learned and who they are learned from. This filter says match any route coming in any interface, per the PERMIT_ALL list, and allow them to come in as long as they were not learned from R4, per the **deny `155.1.0.4/32`** syntax.

☠ Pitfall

Routing filters introduce the possibility of routing loops due to route feedback, or traffic black holes due to incomplete routing information. In this particular scenario route feedback is introduced on the segments between R4 and R5 with the two summaries R5 learns from R4.

Without our filtering configuration R5 learns the 30.0.0.0/14 and 31.0.0.0/14 summaries from both links to R4. Since split-horizon is enabled on the point-to-point link to R4, this information is not sent back out this link. However since split-horizon was disabled on the Frame Relay link these two routes are sent back out this link. When this update is generated, the next-hop value of the originating router on the subnet is used. In this case the value of the next-hop field is 155.1.0.4 for R4. This can be seen from the routing output on R1, R2, or R3:

```
Rack1R1#show ip route 30.0.0.1
Routing entry for 30.0.0.0/14
  Known via "rip", distance 120, metric 3
  Redistributing via rip
  Last update from 155.1.0.4 on Serial0/0.1, 00:00:00 ago
  Routing Descriptor Blocks:
  * 155.1.0.4, from 155.1.0.5, 00:00:00 ago, via Serial0/0.1
      Route metric is 3, traffic share count is 1
```

When R4 receives this update back in is it discarded, because its own local IP address is the next-hop value in the update. The problem in this scenario comes in when the update R5 receives from R4 in the Frame Relay link is filtered out. In this case, R5 receives two updates, one from 155.1.0.4 and one from 155.1.45.4. The update from 155.1.0.4 is filtered out, per the distribute-list gateway filter, and the route via 155.1.45.4 is installed. This route is then sent out the Frame Relay link to all neighbors on the segment, with R5 set as the next-hop.

When R4 receives this update back in the Frame Relay link it is installed as a valid route in the routing table. This is due to the fact that **RIP does not generate a route to Null0 when generating summaries.** Shortly after being installed R4 realizes that the route is not valid, since it is locally generating the summary, and the route is poisoned. This process results in an infinite loop of the route being advertised and withdrawn. The result of this can be seen on R2 as a periodic traffic drop.

```
Rack1R2#ping 30.0.0.1 repeat 100

Type escape sequence to abort.
Sending 100, 100-byte ICMP Echos to 30.0.0.1, timeout is 2 seconds:
..!!!!!U.....!!!!!U.U....!U.
Success rate is 39 percent (11/28), round-trip min/avg/max = 100/102/105 ms
```

The step by step troubleshooting and verification of this problem is as follows. Note that other interfaces in the topology connecting to R4 and R5 have been disabled to simplify debug output, NTP is configured, and timestamping to the millisecond has been enabled to correlate events on both devices simultaneously.

To start, R4 sends the initial update containing 30.0.0.0/14 and 31.0.0.0/14 to R5.

```
Rack1R4#
Jun 26 13:44:01.349: RIP: sending v2 update to 224.0.0.9 via Serial0/1
(155.1.45.4)
Jun 26 13:44:01.349: RIP: build update entries
Jun 26 13:44:01.349:    30.0.0.0/14 via 0.0.0.0, metric 2, tag 0
Jun 26 13:44:01.349:    31.0.0.0/14 via 0.0.0.0, metric 2, tag 0
Jun 26 13:44:01.349:    150.1.4.0/24 via 0.0.0.0, metric 1, tag 0
Jun 26 13:44:01.349:    155.1.0.0/24 via 0.0.0.0, metric 1, tag 0
Jun 26 13:44:01.353:    204.12.1.0/24 via 0.0.0.0, metric 1, tag 0
```

R5 receives this update and installs it in the RIP database and the routing table.

```
Rack1R5#
Jun 26 13:44:01.373: RIP: received v2 update from 155.1.45.4 on Serial0/1
Jun 26 13:44:01.373:      30.0.0.0/14 via 0.0.0.0 in 2 hops
Jun 26 13:44:01.373: RT: SET_LAST_RDB for 30.0.0.0/14
  NEW rdb: via 155.1.45.4

Jun 26 13:44:01.373: RT: add 30.0.0.0/14 via 155.1.45.4, rip metric [120/2]
Jun 26 13:44:01.373: RT: NET-RED 30.0.0.0/14
Jun 26 13:44:01.373: RIP-DB: network_update with 30.0.0.0/14 succeeds
Jun 26 13:44:01.377: RIP-DB: adding 30.0.0.0/14 (metric 2) via 155.1.45.4 on
Serial0/1 to RIP database
Jun 26 13:44:01.377: RIP-DB: add 30.0.0.0/14 (metric 2) via 155.1.45.4 on
Serial0/1
Jun 26 13:44:01.377:      31.0.0.0/14 via 0.0.0.0 in 2 hops
Jun 26 13:44:01.377: RT: SET_LAST_RDB for 31.0.0.0/14
  NEW rdb: via 155.1.45.4

Jun 26 13:44:01.381: RT: add 31.0.0.0/14 via 155.1.45.4, rip metric [120/2]
Jun 26 13:44:01.381: RT: NET-RED 31.0.0.0/14
Jun 26 13:44:01.381: RIP-DB: network_update with 31.0.0.0/14 succeeds
Jun 26 13:44:01.381: RIP-DB: adding 31.0.0.0/14 (metric 2) via 155.1.45.4 on
Serial0/1 to RIP database
Jun 26 13:44:01.381: RIP-DB: add 31.0.0.0/14 (metric 2) via 155.1.45.4 on
Serial0/1
Jun 26 13:44:01.381:      150.1.4.0/24 via 0.0.0.0 in 1 hops
```


R5 takes the update from Serial0/1 and sends it out Serial0/0.1

```
Rack1R5#
Jun 26 13:44:02.026: RIP: sending v2 update to 224.0.0.9 via Serial0/0.1
(155.1.0.5)
Jun 26 13:44:02.026: RIP: build update entries
Jun 26 13:44:02.026:    30.0.0.0/14 via 0.0.0.0, metric 3, tag 0
Jun 26 13:44:02.026:    31.0.0.0/14 via 0.0.0.0, metric 3, tag 0
Jun 26 13:44:02.026:    150.1.4.0/24 via 0.0.0.0, metric 2, tag 0
Jun 26 13:44:02.026:    155.1.0.0/24 via 0.0.0.0, metric 1, tag 0
Jun 26 13:44:02.030:    155.1.45.0/24 via 0.0.0.0, metric 1, tag 0
Jun 26 13:44:02.030:    204.12.1.0/24 via 0.0.0.0, metric 2, tag 0
```

R4 receives the update on Serial0/0.1 from R5. Since there is no Null0 route for the summary R4 mistakenly installs its own origination.

```
Rack1R4#
Jun 26 13:44:02.118: RIP: received v2 update from 155.1.0.5 on Serial0/0.1
Jun 26 13:44:02.118:      30.0.0.0/14 via 0.0.0.0 in 3 hops
Jun 26 13:44:02.118: RT: network 30.0.0.0 is now variably masked
Jun 26 13:44:02.118: RT: SET_LAST_RDB for 30.0.0.0/14
  NEW rdb: via 155.1.0.5

Jun 26 13:44:02.118: RT: add 30.0.0.0/14 via 155.1.0.5, rip metric [120/3]
Jun 26 13:44:02.122: RT: NET-RED 30.0.0.0/14
Jun 26 13:44:02.122: RIP-DB: network_update with 30.0.0.0/14 succeeds
Jun 26 13:44:02.122: RIP-DB: adding 30.0.0.0/14 (metric 3) via 155.1.0.5 on
Serial0/0.1 to RIP database
Jun 26 13:44:02.122: RIP-DB: add 30.0.0.0/14 (metric 3) via 155.1.0.5 on
Serial0/0.1
Jun 26 13:44:02.122:      31.0.0.0/14 via 0.0.0.0 in 3 hops
Jun 26 13:44:02.126: RT: network 31.0.0.0 is now variably masked
Jun 26 13:44:02.126: RT: SET_LAST_RDB for 31.0.0.0/14
  NEW rdb: via 155.1.0.5

Jun 26 13:44:02.126: RT: add 31.0.0.0/14 via 155.1.0.5, rip metric [120/3]
Jun 26 13:44:02.126: RT: NET-RED 31.0.0.0/14
Jun 26 13:44:02.126: RIP-DB: network_update with 31.0.0.0/14 succeeds
Jun 26 13:44:02.130: RIP-DB: adding 31.0.0.0/14 (metric 3) via 155.1.0.5 on
Serial0/0.1 to RIP database
Jun 26 13:44:02.130: RIP-DB: add 31.0.0.0/14 (metric 3) via 155.1.0.5 on
Serial0/0.1
```

Two seconds later R4 realizes that the route is not valid, and poisons it back to R5 with a metric of 16.

```
Jun 26 13:44:04.125: RIP: sending v2 flash update to 224.0.0.9 via Serial0/0.1
(155.1.0.4)
Jun 26 13:44:04.130: RIP: build flash update entries
Jun 26 13:44:04.130:    30.0.0.0/14 via 155.1.0.5, metric 16, tag 0
Jun 26 13:44:04.130:    31.0.0.0/14 via 155.1.0.5, metric 16, tag 0
Jun 26 13:44:04.130: RIP: sending v2 flash update to 224.0.0.9 via Serial0/1
(155.1.45.4)
Jun 26 13:44:04.130: RIP: build flash update entries
Jun 26 13:44:04.130:    30.0.0.0/14 via 0.0.0.0, metric 16, tag 0
Jun 26 13:44:04.134:    31.0.0.0/14 via 0.0.0.0, metric 16, tag 0
```

R5 receives the invalid route from R4, withdraws the route from the routing table and database, and poisons the route back to R4.

```
Jun 26 13:44:04.149: RIP: received v2 update from 155.1.45.4 on Serial0/1
Jun 26 13:44:04.149:      30.0.0.0/14 via 0.0.0.0 in 16 hops  (inaccessible)
Jun 26 13:44:04.149: RT: del 30.0.0.0/14 via 155.1.45.4, rip metric [120/2]
Jun 26 13:44:04.149: RT: delete subnet route to 30.0.0.0/14
Jun 26 13:44:04.149: RT: NET-RED 30.0.0.0/14
Jun 26 13:44:04.153: RT: delete network route to 30.0.0.0
Jun 26 13:44:04.153: RT: NET-RED 30.0.0.0/8
Jun 26 13:44:04.153: RIP-DB: Remove 30.0.0.0/14, (metric 4294967295) via
155.1.45.4, Serial0/1
Jun 26 13:44:04.153:      31.0.0.0/14 via 0.0.0.0 in 16 hops  (inaccessible)
Jun 26 13:44:04.153: RT: del 31.0.0.0/14 via 155.1.45.4, rip metric [120/2]
Jun 26 13:44:04.157: RT: delete subnet route to 31.0.0.0/14
Jun 26 13:44:04.157: RT: NET-RED 31.0.0.0/14
Jun 26 13:44:04.157: RT: delete network route to 31.0.0.0
Jun 26 13:44:04.157: RT: NET-RED 31.0.0.0/8
Jun 26 13:44:04.157: RIP-DB: Remove 31.0.0.0/14, (metric 4294967295) via
155.1.45.4, Serial0/1
Jun 26 13:44:04.217: RIP: received v2 update from 155.1.0.4 on Serial0/0.1
Jun 26 13:44:06.156: RIP: sending v2 flash update to 224.0.0.9 via Serial0/0.1
(155.1.0.5)
Jun 26 13:44:06.156: RIP: build flash update entries
Jun 26 13:44:06.156:    30.0.0.0/14 via 0.0.0.0, metric 16, tag 0
Jun 26 13:44:06.156:    31.0.0.0/14 via 0.0.0.0, metric 16, tag 0
Jun 26 13:44:06.156: RIP: sending v2 flash update to 224.0.0.9 via Serial0/1
(155.1.45.5)
Jun 26 13:44:06.156: RIP: build flash update entries
Jun 26 13:44:06.160:    30.0.0.0/14 via 0.0.0.0, metric 16, tag 0
Jun 26 13:44:06.160:    31.0.0.0/14 via 0.0.0.0, metric 16, tag 0
```

R4 receives the invalid route back from R5 again, and likewise withdraws the route from the routing table and database. The process then continues indefinitely.

```
Jun 26 13:44:06.173: RIP: received v2 update from 155.1.45.5 on Serial0/1
Jun 26 13:44:06.177:      30.0.0.0/14 via 0.0.0.0 in 16 hops  (inaccessible)
Jun 26 13:44:06.177:      31.0.0.0/14 via 0.0.0.0 in 16 hops  (inaccessible)
Jun 26 13:44:06.281: RIP: received v2 update from 155.1.0.5 on Serial0/0.1
Jun 26 13:44:06.285:      30.0.0.0/14 via 0.0.0.0 in 16 hops  (inaccessible)
Jun 26 13:44:06.285: RT: del 30.0.0.0/14 via 155.1.0.5, rip metric [120/3]
Jun 26 13:44:06.285: RT: delete subnet route to 30.0.0.0/14
Jun 26 13:44:06.285: RT: NET-RED 30.0.0.0/14
Jun 26 13:44:06.285: RIP-DB: Remove 30.0.0.0/14, (metric 4294967295) via
155.1.0.5, Serial0/0.1
Jun 26 13:44:06.289:      31.0.0.0/14 via 0.0.0.0 in 16 hops  (inaccessible)
Jun 26 13:44:06.289: RT: del 31.0.0.0/14 via 155.1.0.5, rip metric [120/3]
Jun 26 13:44:06.289: RT: delete subnet route to 31.0.0.0/14
Jun 26 13:44:06.289: RT: NET-RED 31.0.0.0/14
Jun 26 13:44:06.289: RIP-DB: Remove 31.0.0.0/14, (metric 4294967295) via
155.1.0.5, Serial0/0.1
```

The fix for this problem is to ensure that R4 does not install its own summary back into the routing table. OSPF, EIGRP, and BGP already prevent this problem by design through the usage of a route to Null0 being generated every time a summary is created. Multiple solutions are valid to solve this problem, as long as R4 does not install the route.

In the above solution the fix is to configure static routes to Null0, mimicking the behavior of the other protocols.

Another solution would be to configure distribute-list filtering inbound on R4 to prevent the route from being installed.

Another valid solution would be to offset the route to 16 out on R5, or enable split-horizon, however these solutions would stop the routes from being advertised to the rest of the neighbors on the Frame Relay cloud.

```
Rack1R4#conf t
Enter configuration commands, one per line.  End with CNTL/Z.
Rack1R4(config)#ip route 30.0.0.0 255.252.0.0 Null0
Rack1R4(config)#ip route 31.0.0.0 255.252.0.0 Null0
Rack1R4(config)#end
Rack1R4#

Rack1R2#ping 30.0.0.1 repeat 100

Type escape sequence to abort.
Sending 100, 100-byte ICMP Echos to 30.0.0.1, timeout is 2 seconds:
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
Success rate is 100 percent (100/100), round-trip min/avg/max =
100/112/341 ms
```

4.11 RIPv2 Filtering with Standard Access-Lists

- Configure a one line standard access-list on R6 to filter out all routes coming from BB1 that have an even number in the third octet.

Configuration

```
R6:
router rip
 distribute-list 1 in Serial0/0
!
access-list 1 permit 0.0.1.0 255.255.254.255
```

Verification

```
Rack1R6#show ip route | include Serial0/0
C       54.1.1.0 is directly connected, Serial0/0
R    212.18.1.0/24 [120/1] via 54.1.1.254, 00:00:24, Serial0/0
R    212.18.3.0/24 [120/1] via 54.1.1.254, 00:00:24, Serial0/0
```


4.12 RIPv2 Filtering with Extended Access-Lists

- Remove the previous prefix-list filters on R5.
- Configure an extended access-list filter on R5 so that the routes to VLANs 7 and 9 are only received from R1, while the routes to R1's Loopback and VLAN 146 are only received from R3.
- This filter should not affect any other updates on this segment.

Configuration

```
R5:
router rip
 distribute-list 100 in Serial0/0.1
!
access-list 100 deny   ip host 155.1.0.3 host 155.1.7.0
access-list 100 deny   ip host 155.1.0.3 host 155.1.9.0
access-list 100 deny   ip host 155.1.0.1 host 155.1.146.0
access-list 100 deny   ip host 155.1.0.1 host 150.1.1.0
access-list 100 permit ip any any
```

Verification

Note

Extended access-lists when called as a distribute-list in IGP have a different meaning than in redistribution or as in BGP. With BGP and redistribution the "source" field in the ACL represents the network address, and the "destination" field represents the subnet mask. In IGP distribute-list application the "source" field in the ACL matches the update source of the route, and the "destination" field represents the network address. This implementation allows us to control which networks we are receiving, but more importantly who we are receiving them from. Before the filter is applied, R5 routes to R3 for VLANs 7 and 9, and to R1 for VLAN 146 and R1's Loopback0.

```
Rack1R5#show ip route rip | include via 155.1.0.(1|3)
R       155.1.146.0 [120/1] via 155.1.0.1, 00:00:04, Serial0/0.1
R       155.1.9.0 [120/3] via 155.1.0.3, 00:00:04, Serial0/0.1
R       155.1.13.0 [120/1] via 155.1.0.3, 00:00:04, Serial0/0.1
                   [120/1] via 155.1.0.1, 00:00:04, Serial0/0.1
R       155.1.7.0 [120/2] via 155.1.0.3, 00:00:04, Serial0/0.1
R       155.1.37.0 [120/1] via 155.1.0.3, 00:00:04, Serial0/0.1
R       155.1.79.0 [120/2] via 155.1.0.3, 00:00:04, Serial0/0.1
R       155.1.67.0 [120/2] via 155.1.0.3, 00:00:04, Serial0/0.1
R       54.1.1.0 [120/3] via 155.1.0.3, 00:00:04, Serial0/0.1
R    212.18.1.0/24 [120/4] via 155.1.0.3, 00:00:04, Serial0/0.1
R    212.18.3.0/24 [120/4] via 155.1.0.3, 00:00:04, Serial0/0.1
R       150.1.7.0 [120/2] via 155.1.0.3, 00:00:04, Serial0/0.1
R       150.1.6.0 [120/3] via 155.1.0.3, 00:00:04, Serial0/0.1
R       150.1.3.0 [120/1] via 155.1.0.3, 00:00:04, Serial0/0.1
R       150.1.1.0 [120/1] via 155.1.0.1, 00:00:04, Serial0/0.1
R       150.1.9.0 [120/3] via 155.1.0.3, 00:00:04, Serial0/0.1
```

After the filter is applied, R5 routes to R3 for VLANs 7 and 9, and to R1 for VLAN 146 and R1’s Loopback0.

```
Rack1R5#show ip route rip | include via 155.1.0.(1|3)
R       155.1.146.0 [120/2] via 155.1.0.3, 00:00:02, Serial0/0.1
R       155.1.9.0 [120/4] via 155.1.0.1, 00:00:02, Serial0/0.1
R       155.1.13.0 [120/1] via 155.1.0.3, 00:00:02, Serial0/0.1
                   [120/1] via 155.1.0.1, 00:00:02, Serial0/0.1
R       155.1.7.0 [120/3] via 155.1.0.1, 00:00:02, Serial0/0.1
R       155.1.37.0 [120/1] via 155.1.0.3, 00:00:02, Serial0/0.1
R       155.1.79.0 [120/2] via 155.1.0.3, 00:00:02, Serial0/0.1
R       155.1.67.0 [120/2] via 155.1.0.3, 00:00:02, Serial0/0.1
R       54.1.1.0 [120/3] via 155.1.0.3, 00:00:02, Serial0/0.1
R    212.18.1.0/24 [120/4] via 155.1.0.3, 00:00:02, Serial0/0.1
R    212.18.3.0/24 [120/4] via 155.1.0.3, 00:00:02, Serial0/0.1
R       150.1.7.0 [120/2] via 155.1.0.3, 00:00:02, Serial0/0.1
R       150.1.6.0 [120/3] via 155.1.0.3, 00:00:02, Serial0/0.1
R       150.1.3.0 [120/1] via 155.1.0.3, 00:00:02, Serial0/0.1
R       150.1.1.0 [120/2] via 155.1.0.3, 00:00:02, Serial0/0.1
R       150.1.9.0 [120/3] via 155.1.0.3, 00:00:02, Serial0/0.1
```

4.13 RIPv2 Filtering with Offset Lists

- Configure an offset-list on SW1 so that SW3 does not install a route to VLAN 5.
- This filter should not affect any other updates on this segment.

Configuration

```
SW1:
router rip
 offset-list 1 out 16 Vlan79
!
access-list 1 permit 155.1.5.0
```

Verification

Note

Before offset-list is applied.

```
Rack1SW3#show ip route 155.1.5.0
Routing entry for 155.1.5.0/24
  Known via "rip", distance 120, metric 3
  Redistributing via rip
  Last update from 155.1.79.7 on Vlan79, 00:00:05 ago
  Routing Descriptor Blocks:
  * 155.1.79.7, from 155.1.79.7, 00:00:05 ago, via Vlan79
      Route metric is 3, traffic share count is 1
```

After offset-list is applied.

```
Rack1SW3#show ip route 155.1.5.0
% Subnet not in table

Rack1SW3#debug ip rip
RIP protocol debugging is on
Rack1SW3#
RIP: received v2 update from 155.1.79.7 on Vlan79
     30.0.0.0/14 via 0.0.0.0 in 5 hops
     31.0.0.0/14 via 0.0.0.0 in 5 hops
     54.1.1.0/24 via 0.0.0.0 in 2 hops
     150.1.1.0/24 via 0.0.0.0 in 3 hops
     150.1.2.0/24 via 0.0.0.0 in 4 hops
     150.1.3.0/24 via 0.0.0.0 in 2 hops
     150.1.4.0/24 via 0.0.0.0 in 4 hops
     150.1.5.0/24 via 0.0.0.0 in 3 hops
     150.1.6.0/24 via 0.0.0.0 in 2 hops
     150.1.7.0/24 via 0.0.0.0 in 1 hops
     150.1.8.0/24 via 0.0.0.0 in 4 hops
     155.1.0.0/24 via 0.0.0.0 in 2 hops
     155.1.5.0/24 via 0.0.0.0 in 16 hops  (inaccessible)
     155.1.7.0/24 via 0.0.0.0 in 1 hops
     155.1.8.0/24 via 0.0.0.0 in 4 hops
     155.1.10.0/24 via 0.0.0.0 in 5 hops
     155.1.13.0/24 via 0.0.0.0 in 2 hops
     155.1.37.0/24 via 0.0.0.0 in 1 hops
     155.1.45.0/24 via 0.0.0.0 in 3 hops
     155.1.58.0/24 via 0.0.0.0 in 3 hops
     155.1.67.0/24 via 0.0.0.0 in 1 hops
     155.1.108.0/24 via 0.0.0.0 in 4 hops
     155.1.146.0/24 via 0.0.0.0 in 2 hops
```

4.14 RIPv2 Filtering with Administrative Distance

- Configure administrative distance filtering on R5 so that devices within the network cannot reach R4's Loopback0 network.
- This filter should not affect any other networks in the topology.

Configuration

```
R5:
router rip
 distance 255 0.0.0.0 255.255.255.255 1
!
access-list 1 permit 150.1.4.0
```

Verification

Note

Before distance is applied.

```
Rack1R5#show ip route 150.1.4.0
Routing entry for 150.1.4.0/24
  Known via "rip", distance 120, metric 1
  Redistributing via rip
  Last update from 155.1.45.4 on Serial0/1, 00:00:01 ago
  Routing Descriptor Blocks:
  * 155.1.45.4, from 155.1.45.4, 00:00:01 ago, via Serial0/1
      Route metric is 1, traffic share count is 1
    155.1.0.4, from 155.1.0.4, 00:00:07 ago, via Serial0/0.1
      Route metric is 1, traffic share count is 1

Rack1R2#ping 150.1.4.4

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 150.1.4.4, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 112/116/124
ms
```

Even though administrative distance is locally significant to the router, the RIP process, like the EIGRP process, cannot advertise a route that is not actually installed in the routing table. By setting the distance of the route 150.1.4.0 to 255, it is invalidated from being installed in the routing table, and hence invalidated from being advertised to any neighbors. Once complete R2 not only does not have reachability to the destination, but it does not have the route in its routing table or RIP database.

```
Rack1R5#show ip route 150.1.4.0
% Subnet not in table

Rack1R2#ping 150.1.4.4

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 150.1.4.4, timeout is 2 seconds:
.....
Success rate is 0 percent (0/5)

Rack1R2#show ip route 150.1.4.0
% Subnet not in table

Rack1R2#show ip rip database 150.1.4.0 255.255.255.0
%Route not in database
```

4.15 RIPv2 Filtering with Per Neighbor AD

- Configure administrative distance filtering on SW1 so that traffic destined for R3's Loopback0 network is sent towards R6.
- This configuration should not affect any other networks in the topology.

Configuration

```
SW1:
router rip
 distance 255 155.1.37.3 0.0.0.0 2
!
access-list 2 permit 150.1.3.0
```

Verification

> **Note**
>
> Before distance filtering.

```
Rack1SW1#show ip route 150.1.3.0
Routing entry for 150.1.3.0/24
  Known via "rip", distance 120, metric 1
  Redistributing via rip
  Last update from 155.1.37.3 on FastEthernet0/3, 00:00:01 ago
  Routing Descriptor Blocks:
  * 155.1.37.3, from 155.1.37.3, 00:00:01 ago, via FastEthernet0/3
      Route metric is 1, traffic share count is 1
```

> The distance command can be used to change the administrative distance globally for the routing process, globally for the routing process per route type (i.e. external vs. internal), on a per-prefix basis, or on a per-neighbor per-prefix basis. The two fields after the distance value are the source of the route and a wildcard mask to match the source. The value needed for the source is seen in the above output as the **`from 155.1.37.3`** field.

```
Rack1SW1#show ip route 150.1.3.0
Routing entry for 150.1.3.0/24
  Known via "rip", distance 120, metric 3
  Redistributing via rip
  Last update from 155.1.67.6 on Vlan67, 00:00:05 ago
  Routing Descriptor Blocks:
  * 155.1.67.6, from 155.1.67.6, 00:00:05 ago, via Vlan67
      Route metric is 3, traffic share count is 1
```

4.16 RIPv2 Default Routing

- Configure R6 to advertise a default route only to R1 via RIP.
- R6 should not send this default route directly to SW1.
- Do not use any access-list or prefix-lists on R6 to accomplish this.

Configuration

```
R6:
router rip
 default-information originate route-map DEFAULT_TO_R1
!
route-map DEFAULT_TO_R1 permit 10
 set interface FastEthernet0/0.146

SW1:
router rip
 distribute-list prefix NO_DEFAULT out Vlan67
!
ip prefix-list NO_DEFAULT seq 5 deny 0.0.0.0/0
ip prefix-list NO_DEFAULT seq 10 permit 0.0.0.0/0 le 32
```

Verification

```
Rack1R6#show ip rip database 0.0.0.0 0.0.0.0
0.0.0.0/0    redistributed
    [1] via 0.0.0.0,

Rack1R1#show ip route | include ( 0.0.0.0)
Gateway of last resort is 155.1.146.6 to network 0.0.0.0
R*   0.0.0.0/0 [120/6] via 155.1.146.6, 00:00:06, FastEthernet0/0

Rack1R3#show ip route | include ( 0.0.0.0)
Gateway of last resort is 155.1.13.1 to network 0.0.0.0
R*   0.0.0.0/0 [120/7] via 155.1.13.1, 00:00:06, Serial1/2

Rack1R6#show ip route | include ( 0.0.0.0)

Rack1SW1#show ip route | include ( 0.0.0.0)
Gateway of last resort is 155.1.37.3 to network 0.0.0.0
R*   0.0.0.0/0 [120/8] via 155.1.37.3, 00:00:00, FastEthernet0/3
```

☠ Pitfall

Note in the above output that R6 does not have a default route installed in the routing table. Unlike OSPF, RIP does not require that a default route actually be installed in the routing table before originating one. Similarly to the previous problem in this topology with the route feedback of R4's summary route, route feedback of R6's default origination will occur in this topology.

Due to the route-map attached to the default origination of R6, the default advertisement is only sent out VLAN 146 towards R1. When R1 receives this it is installed in the routing table and advertised on to R3. Note that the metric of the default route is 6 when it reaches R1 due to the previous offset-list with a metric of 5 configured outbound on R6's link to VLAN 146.

```
Rack1R1#debug ip rip
RIP protocol debugging is on
RIP: received packet with text authentication CCIE
RIP: received v2 update from 155.1.146.6 on FastEthernet0/0
     0.0.0.0/0 via 0.0.0.0 in 6 hops
<output omitted>
RIP: sending v2 update to 224.0.0.9 via Serial0/1 (155.1.13.1)
RIP: build update entries
        0.0.0.0/0 via 0.0.0.0, metric 7, tag 0
<output omitted>
```

R3 now receives the update in from R1 with a metric of 7, and forwards the announcement to SW1 with a metric of 8.

```
Rack1R3#debug ip rip
RIP protocol debugging is on
RIP: received v2 update from 155.1.13.1 on Serial1/2
     0.0.0.0/0 via 0.0.0.0 in 7 hops
<output omitted>
RIP: sending v2 update to 224.0.0.9 via FastEthernet0/0 (155.1.37.3)
RIP: build update entries
        0.0.0.0/0 via 0.0.0.0, metric 8, tag 0
<output omitted>
```

SW1 receives the default route from R3 with a metric of 8 and installs it. Normally SW1 would continue to advertise this route on to R6, but in the above example the default route is filtered out from this advertisement with a prefix-list applied as a distribute-list.

```
Rack1SW1#debug ip rip
RIP protocol debugging is on
RIP: received v2 update from 155.1.37.3 on FastEthernet0/3
     0.0.0.0/0 via 0.0.0.0 in 8 hops
<output omitted>
RIP: sending v2 update to 224.0.0.9 via Vlan67 (155.1.67.7)
RIP: build update entries
        30.0.0.0/14 via 0.0.0.0, metric 5, tag 0
        31.0.0.0/14 via 0.0.0.0, metric 5, tag 0
        150.1.2.0/24 via 0.0.0.0, metric 4, tag 0
        150.1.5.0/24 via 0.0.0.0, metric 3, tag 0
        150.1.7.0/24 via 0.0.0.0, metric 1, tag 0
        150.1.8.0/24 via 0.0.0.0, metric 4, tag 0
        150.1.9.0/24 via 0.0.0.0, metric 2, tag 0
        155.1.0.0/24 via 0.0.0.0, metric 2, tag 0
```

```
        155.1.5.0/24 via 0.0.0.0, metric 3, tag 0
        155.1.7.0/24 via 0.0.0.0, metric 1, tag 0
        155.1.8.0/24 via 0.0.0.0, metric 4, tag 0
        155.1.9.0/24 via 0.0.0.0, metric 2, tag 0
        155.1.10.0/24 via 0.0.0.0, metric 5, tag 0
        155.1.13.0/24 via 0.0.0.0, metric 2, tag 0
        155.1.37.0/24 via 0.0.0.0, metric 1, tag 0
        155.1.45.0/24 via 0.0.0.0, metric 3, tag 0
        155.1.58.0/24 via 0.0.0.0, metric 3, tag 0
        155.1.79.0/24 via 0.0.0.0, metric 1, tag 0
        155.1.108.0/24 via 0.0.0.0, metric 4, tag 0
        192.10.1.0/24 via 0.0.0.0, metric 4, tag 0
        204.12.1.0/24 via 0.0.0.0, metric 4, tag 0
        205.90.31.0/24 via 0.0.0.0, metric 11, tag 0
        220.20.3.0/24 via 0.0.0.0, metric 11, tag 0
        222.22.2.0/24 via 0.0.0.0, metric 11, tag 0
```

Since R6 does not receive the default route from SW1 there is no feedback. Now let's look at what happens in the topology when there is no filter configured on SW1.

```
Rack1SW1#conf t
Enter configuration commands, one per line.  End with CNTL/Z.
Rack1SW1(config)#router rip
Rack1SW1(config-router)#no distribute-list prefix NO_DEFAULT out Vlan67
Rack1SW1(config-router)#end
Rack1SW1#
```

R6 originates the default route to R1 with a metric of 6.

```
Rack1R6#debug ip rip
RIP protocol debugging is on
Rack1R6#debug ip routing
IP routing debugging is on

Jun 27 05:19:03.009: RIP: sending v2 update to 224.0.0.9 via
FastEthernet0/0.146 (155.1.146.6)
Jun 27 05:19:03.009: RIP: build update entries
Jun 27 05:19:03.009:     0.0.0.0/0 via 0.0.0.0, metric 6, tag 0
```

The route is sent from R1, to R3, to SW1, and then fed back to R6. Since R6 does not actually have the default installed in the routing table it thinks that SW1's advertisement is valid.

```
Jun 27 05:19:09.059: RIP: received v2 update from 155.1.67.7 on
FastEthernet0/0.67
Jun 27 05:19:09.059:      0.0.0.0/0 via 0.0.0.0 in 9 hops
Jun 27 05:19:09.059: RT: SET_LAST_RDB for 0.0.0.0/0
  NEW rdb: via 155.1.67.7

Jun 27 05:19:09.059: RT: add 0.0.0.0/0 via 155.1.67.7, rip metric
[120/9]
Jun 27 05:19:09.063: RT: NET-RED 0.0.0.0/0
Jun 27 05:19:09.067: RT: default path is now 0.0.0.0 via 155.1.67.7
Jun 27 05:19:09.067: RT: new default network 0.0.0.0
Jun 27 05:19:09.067: RT: NET-RED 0.0.0.0/0
Jun 27 05:19:09.203: RIP: received v2 update from 155.1.67.7 on
FastEthernet0/0.67
Jun 27 05:19:09.203:      0.0.0.0/0 via 0.0.0.0 in 9 hops
```

R6 now sends a triggered update to R1 with the new default route. The metric is incremented to 10, and then offset 5 for a total of 15 as the route is sent to R1. Even without the offset list on R6 the problem will still occur, however the offset list makes the loop occur more quickly as R6 is feeding the route back with metric increments of 5 instead of 1.

```
Jun 27 05:19:11.070: RIP: sending v2 flash update to 224.0.0.9 via
FastEthernet0/0.146 (155.1.146.6)
Jun 27 05:19:11.074: RIP: build flash update entries
Jun 27 05:19:11.074:    0.0.0.0/0 via 0.0.0.0, metric 15, tag 0
```

The result of this problem can also be viewed in the rest of the topology through the **debug ip routing** output.

```
Rack1R1#debug ip routing
IP routing debugging is on
RT: SET_LAST_RDB for 0.0.0.0/0
  NEW rdb: via 155.1.146.6

RT: add 0.0.0.0/0 via 155.1.146.6, rip metric [120/6]
RT: NET-RED 0.0.0.0/0
RT: default path is now 0.0.0.0 via 155.1.146.6
RT: new default network 0.0.0.0
RT: NET-RED 0.0.0.0/0
RT: rip's 0.0.0.0/0 (via 155.1.146.6) metric changed from
distance/metric [120/6] to [120/15]
RT: NET-RED 0.0.0.0/0
RT: del 0.0.0.0 via 155.1.146.6, rip metric [120/15]
RT: delete network route to 0.0.0.0
RT: NET-RED 0.0.0.0/0
```

```
Rack1R3#debug ip routing
IP routing debugging is on
RT: SET_LAST_RDB for 0.0.0.0/0
  NEW rdb: via 155.1.13.1

RT: add 0.0.0.0/0 via 155.1.13.1, rip metric [120/7]
RT: NET-RED 0.0.0.0/0
RT: default path is now 0.0.0.0 via 155.1.13.1
RT: new default network 0.0.0.0
RT: NET-RED 0.0.0.0/0
RT: NET-RED 0.0.0.0/0
RT: del 0.0.0.0 via 155.1.13.1, rip metric [120/7]
RT: delete network route to 0.0.0.0
RT: NET-RED 0.0.0.0/0
RT: NET-RED 0.0.0.0/0
```

```
Rack1SW1#debug ip routing
IP routing debugging is on
RT: SET_LAST_RDB for 0.0.0.0/0
  NEW rdb: via 155.1.37.3

RT: add 0.0.0.0/0 via 155.1.37.3, rip metric [120/8]
RT: default path is now 0.0.0.0 via 155.1.37.3
RT: new default network 0.0.0.0
RT: del 0.0.0.0 via 155.1.37.3, rip metric [120/8]
RT: delete network route to 0.0.0.0
```

To fix this problem we need to ensure that R6 does not install a default route via SW1. One way to fix this, as seen above, is to filter the default route from being sent from SW1 to R6. Another solution would be to filter the default in on R6 from SW1. Yet another solution would be to actually configure a static default route on R6, either to Null0 or to another interface, so the RIP route could not be installed in the routing table from SW1.

The simplest solution, although it does not meet the task requirements, is to have R6 advertise the default route out all interfaces. If R6 sends the default route directly to SW1, SW1 cannot send it back to R6 due to split-horizon, and the loop is avoided.

4.17 RIPv2 Conditional Default Routing

- Remove the previous default route advertisement on R6.
- Configure R4 to originate a default route into the RIP domain.
- If the link to BB3 goes down R4 should withdraw its default advertisement.

Configuration

```
R4:
router rip
 default-information originate route-map TRACK_LINK_TO_BB3
!
ip prefix-list LINK_TO_BB3 seq 5 permit 204.12.1.0/24
!
route-map TRACK_LINK_TO_BB3 permit 10
 match ip address prefix-list LINK_TO_BB3
```

Verification

Note

As long as R4 has a route to the network 204.12.1.0/24 installed in the routing table it will advertise a default route.

```
Rack1R4#show ip route 204.12.1.0 255.255.255.0
Routing entry for 204.12.1.0/24
  Known via "connected", distance 0, metric 0 (connected, via
interface)
  Redistributing via rip
  Advertised by rip
  Routing Descriptor Blocks:
  * directly connected, via FastEthernet0/0
      Route metric is 0, traffic share count is 1

Rack1R5#show ip route | include ( 0.0.0.0)
Gateway of last resort is 155.1.45.4 to network 0.0.0.0
R*   0.0.0.0/0 [120/1] via 155.1.45.4, 00:00:05, Serial0/1
```


If 204.12.1.0/24 leaves the routing table, the default route is withdrawn.

```
Rack1SW2#conf t
Enter configuration commands, one per line.  End with CNTL/Z.
Rack1SW2(config)#interface Fa0/4
Rack1SW2(config-if)#shutdown
%LINK-5-CHANGED: Interface FastEthernet0/4, changed state to
administratively down
%LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/4,
changed state to down

Rack1R4#show ip route 204.12.1.0 255.255.255.0
% Network not in table

Rack1R5#show ip route | include ( 0.0.0.0)
```

☠ Pitfall

Remember that the link status of a connected interface is not always a good indication of end-to-end reachability on the segment. In this design if the trunk links in the network fail, if the link from SW3 to BB3 fails, or if BB3 itself fails, R4 has no way to know this. Since the installation of the 204.12.1.0/24 route on R4 is solely dependent on the link-local status between R4 and SW2, this is not a truly fault tolerant design.

4.18 RIPv2 Reliable Conditional Default Routing

- Configure the IP SLA feature on R4 to track ICMP reachability to BB3.
- Modify the previous default route origination on R4 so that if an ICMP echo-reply is not received from BB3 R4 withdraws its default advertisement.

Configuration

```
R4:
ip sla monitor 1
 type echo protocol ipIcmpEcho 204.12.1.254 source-interface
FastEthernet0/0
 timeout 50
 frequency 1
ip sla monitor schedule 1 start-time now
!
track 1 rtr 1
!
router rip
 default-information originate route-map RELIABLY_TRACK_LINK_TO_BB3
!
ip route 169.254.0.1 255.255.255.255 Null0 track 1
!
ip prefix-list DUMMY_ROUTE_TRACKED_VIA_SLA seq 5 permit 169.254.0.1/32
!
route-map RELIABLY_TRACK_LINK_TO_BB3 permit 10
 match ip address prefix-list DUMMY_ROUTE_TRACKED_VIA_SLA
```

Verification

Note

In the previous example R4 could not detect an indirect failure in the transit path to BB3. With this example an IP SLA instance is introduced into conditional default origination. The SLA instance checks reachability to BB3 via ICMP every five seconds, with a timeout of two seconds. The SLA instance is then called from an enhanced object, which is called from a static route. This link local route of 169.254.0.1/32 could be any arbitrary dummy prefix. The dummy prefix is then called from a route-map which is tied to the default route origination. Therefore if R4 loses ICMP reachability to BB3 the default route is withdrawn.

With the network properly functioning R4 advertises its default route to R5.

```
Rack1R5#show ip route | in ( 0.0.0.0)
Gateway of last resort is 155.1.45.4 to network 0.0.0.0
R*   0.0.0.0/0 [120/1] via 155.1.45.4, 00:00:06, Serial0/1
```

Next an indirect link failure is simulated by disabling the link between SW3 and BB3.

```
Rack1R4#debug ip sla monitor trace
IP SLA Monitor TRACE debugging for all operation is on

Rack1R4#debug track

Rack1R4#debug ip routing
IP routing debugging is on

Rack1R5#debug ip rip
RIP protocol debugging is on

Rack1R5#debug ip routing
IP routing debugging is on

Rack1SW3#conf t
Enter configuration commands, one per line.  End with CNTL/Z.
Rack1SW3(config)#interface Fa0/24
Rack1SW3(config-if)#shutdown
Rack1SW3(config-if)#
Jun 27 06:44:19.657: %LINK-5-CHANGED: Interface FastEthernet0/24,
changed state to administratively down
Jun 27 06:44:20.657: %LINEPROTO-5-UPDOWN: Line protocol on Interface
FastEthernet0/24, changed state to down
```

SW3 declares the link to BB3 down. R4 immediately detects this and declares its object down.

```
Rack1R4#
Jun 27 06:44:21.092: IP SLA Monitor(1) Scheduler: Starting an operation
Jun 27 06:44:21.092: IP SLA Monitor(1) echo operation: Sending an echo
operation
Jun 27 06:44:21.144: IP SLA Monitor(1) echo operation: Timeout
Jun 27 06:44:21.144: IP SLA Monitor(1) Scheduler: Updating result
Jun 27 06:44:22.094: IP SLA Monitor(1) Scheduler: Starting an operation
Jun 27 06:44:22.094: IP SLA Monitor(1) echo operation: Sending an echo
operation
Jun 27 06:44:22.146: IP SLA Monitor(1) echo operation: Timeout
Jun 27 06:44:22.146: IP SLA Monitor(1) Scheduler: Updating result
Jun 27 06:44:22.406: Track: 1 Change #16 rtr 1, state Up->Down
```

This causes R4 to withdraw the dummy route from the routing table.

```
Rack1R4#
Jun 27 06:44:22.406: RT: del 169.254.0.1/32 via 0.0.0.0, static metric [1/0]
Jun 27 06:44:22.406: RT: delete subnet route to 169.254.0.1/32
```

This causes the route-map condition to fail, which in turn causes R4 to poison its default route advertisement.

```
Rack1R5#
Jun 27 06:44:24.422: RIP: received v2 update from 155.1.45.4 on Serial0/1
Jun 27 06:44:24.422:      0.0.0.0/0 via 0.0.0.0 in 16 hops  (inaccessible)
Jun 27 06:44:24.426: RT: del 0.0.0.0 via 155.1.45.4, rip metric [120/1]
Jun 27 06:44:24.426: RT: SET_LAST_RDB for 0.0.0.0/0
  OLD rdb: via 11.13.11.13
  NEW rdb: via 155.1.0.4, Serial0/0.1

Jun 27 06:44:24.426: RT: NET-RED 0.0.0.0/0
Jun 27 06:44:24.510: RIP: received v2 update from 155.1.0.4 on Serial0/0.1
Jun 27 06:44:24.510:      0.0.0.0/0 via 0.0.0.0 in 16 hops  (inaccessible)
Jun 27 06:44:24.510: RT: del 0.0.0.0 via 155.1.0.4, rip metric [120/1]
Jun 27 06:44:24.510: RT: delete network route to 0.0.0.0
Jun 27 06:44:24.510: RT: NET-RED 0.0.0.0/0
Jun 27 06:44:24.510: RT: NET-RED 0.0.0.0/0
```

4.19 RIPv2 Unicast Updates

- Configure R5 and SW2 so that RIPv2 updates sent over VLAN 58 use unicasts instead of multicasts.

Configuration

```
R5:
router rip
 passive-interface FastEthernet0/0
 neighbor 155.1.58.8

SW2:
router rip
 passive-interface Vlan58
 neighbor 155.1.58.5
```

Verification

> **Note**
>
> Like EIGRP and OSPF the **neighbor** statement in RIP is used to send updates out an interface as unicast. Unlike EIGRP however the **neighbor** statement does not automatically suppress the sending of the broadcast or multicast update. The additional **passive-interface** command is required to accomplish this.

```
Rack1SW2#debug ip packet detail
IP packet debugging is on (detailed)
IP: s=155.1.58.8 (local), d=155.1.58.5 (Vlan58), len 112, sending
    UDP src=520, dst=520
IP: s=155.1.58.8 (local), d=155.1.58.5 (Vlan58), len 112, sending full
packet
    UDP src=520, dst=520
IP: s=155.1.58.5 (Vlan58), d=155.1.58.8, len 532, rcvd 2
    UDP src=520, dst=520
IP: s=155.1.58.5 (Vlan58), d=155.1.58.8, len 532, stop process pak for
forus packet
    UDP src=520, dst=520
```

4.20 RIPv2 Broadcast Updates

- Configure R1 and R6 so that RIPv2 updates sent over VLAN 146 use broadcasts instead of multicasts.

Configuration

```
R1:
interface FastEthernet0/0
 ip rip v2-broadcast

R6:
interface FastEthernet0/0.146
 ip rip v2-broadcast
```

Verification

Note

Normally RIPv2 updates are sent as multicast. The interface level command ip rip v2-broadcast reverts back to using the all host broadcast address 255.255.255.255 for updates.

```
Rack1R1#debug ip packet detail
IP packet debugging is on (detailed)
IP: s=155.1.146.1 (local), d=255.255.255.255 (FastEthernet0/0), len
532, sending broad/multicast
    UDP src=520, dst=520
IP: s=155.1.146.1 (local), d=255.255.255.255 (FastEthernet0/0), len
212, sending broad/multicast
    UDP src=520, dst=520
IP: s=155.1.13.1 (local), d=224.0.0.9 (Serial0/1), len 432, sending
broad/multicast
    UDP src=520, dst=520
IP: s=155.1.146.6 (FastEthernet0/0), d=255.255.255.255, len 272, rcvd 2
    UDP src=520, dst=520
```


4.21 RIPv2 Triggered Updates

- Configure R4 and R5 so that RIPv2 updates are only exchanged over the low speed point-to-point Serial link between them when there is a change in the RIP topology.

Configuration

```
R4:
interface Serial0/1
 ip rip triggered

R5:
interface Serial0/1
 ip rip triggered
```

Verification

Note

RFC 2091 - Triggered Extensions to RIP to Support Demand Circuits, defines a mechanism similar to OSPF demand circuit in which periodic updates are suppressed out a link. Updates are only sent out triggered links when there is a change in the RIP database. The below output indicates that R5 is sending updates out all its interfaces except the Serial0/1 link to R4.

```
Rack1R5#debug ip rip
RIP protocol debugging is on
RIP: sending v2 update to 224.0.0.9 via Serial0/0.1 (155.1.0.5)
RIP: build update entries
        0.0.0.0/0 via 155.1.0.4, metric 2, tag 0
        30.0.0.0/14 via 155.1.0.4, metric 3, tag 0
        31.0.0.0/14 via 155.1.0.4, metric 3, tag 0
<output omitted>
RIP: sending v2 update to 224.0.0.9 via FastEthernet0/1 (155.1.5.5)
RIP: build update entries
        0.0.0.0/0 via 0.0.0.0, metric 2, tag 0
        30.0.0.0/14 via 0.0.0.0, metric 3, tag 0
        31.0.0.0/14 via 0.0.0.0, metric 3, tag 0
<output omitted>
RIP: sending v2 update to 155.1.58.8 via FastEthernet0/0 (155.1.58.5)
RIP: build update entries
        0.0.0.0/0 via 0.0.0.0, metric 2, tag 0
        30.0.0.0/14 via 0.0.0.0, metric 3, tag 0
        31.0.0.0/14 via 0.0.0.0, metric 3, tag 0
<output omitted>
```


4.22 RIPv2 Source Validation

- Configure R1 and R3 to use PPP on the Serial link between them.
- Remove R1's IP address on this segment.
- Configure R3 to assign R1 the IP address 155.1.13.1/32 via IPCP.
- Ensure that RIPv2 updates sent across this link can be installed in the routing tables of these two devices.

Configuration

```
R1:
interface Serial0/1
 ip address negotiated
encapsulation ppp
!
router rip
 no validate-update-source

R3:
interface Serial1/2
 encapsulation ppp
 peer default ip address 155.1.13.1
```

Verification

Note

Prior to IP address modification on R1, RIP routes from R3 are installed.

```
Rack1R1#show ip route | include via 155.1.13.3
R       155.1.9.0 [120/3] via 155.1.13.3, 00:00:03, Serial0/1
R       155.1.7.0 [120/2] via 155.1.13.3, 00:00:03, Serial0/1
R       155.1.37.0 [120/1] via 155.1.13.3, 00:00:03, Serial0/1
R       155.1.79.0 [120/2] via 155.1.13.3, 00:00:03, Serial0/1
R       155.1.67.0 [120/2] via 155.1.13.3, 00:00:03, Serial0/1
R       54.1.1.0 [120/3] via 155.1.13.3, 00:00:03, Serial0/1
R    212.18.1.0/24 [120/4] via 155.1.13.3, 00:00:03, Serial0/1
R    212.18.3.0/24 [120/4] via 155.1.13.3, 00:00:03, Serial0/1
R       150.1.7.0 [120/2] via 155.1.13.3, 00:00:03, Serial0/1
R       150.1.6.0 [120/3] via 155.1.13.3, 00:00:03, Serial0/1
R       150.1.3.0 [120/1] via 155.1.13.3, 00:00:03, Serial0/1
R       150.1.9.0 [120/3] via 155.1.13.3, 00:00:03, Serial0/1
```

After the IP address is negotiated on R1, it installs a peer-neighbor host route to R3 as 155.1.13.3/32. Since the local link, 155.1.13.1/32, and R3's link are not on the same subnet, R1 cannot accept RIP updates from R3. However, R3 can receive updates from R3, because R3 still sees its own interface as 155.1.13.0/24, which 155.1.13.1 is a part of.

```
Rack1R1#show ip interface brief
Interface                  IP-Address      OK? Method Status
Protocol
FastEthernet0/0            155.1.146.1     YES manual up                    up
Serial0/0                  unassigned      YES unset  up                    up
Serial0/0.1                155.1.0.1       YES manual up                    up
Serial0/1                  155.1.13.1      YES IPCP   up                    up
Loopback0                  150.1.1.1       YES manual up                    up

Rack1R1#show ip route connected
     155.1.0.0/16 is variably subnetted, 16 subnets, 2 masks
C       155.1.146.0/24 is directly connected, FastEthernet0/0
C       155.1.13.3/32 is directly connected, Serial0/1
C       155.1.13.1/32 is directly connected, Serial0/1
C       155.1.0.0/24 is directly connected, Serial0/0.1
     150.1.0.0/24 is subnetted, 8 subnets
C       150.1.1.0 is directly connected, Loopback0

Rack1R3#show ip route connected
     155.1.0.0/16 is variably subnetted, 15 subnets, 2 masks
C       155.1.13.1/32 is directly connected, Serial1/2
C       155.1.13.0/24 is directly connected, Serial1/2
C       155.1.0.0/24 is directly connected, Serial1/0.1
C       155.1.37.0/24 is directly connected, FastEthernet0/0
     150.1.0.0/24 is subnetted, 8 subnets
C       150.1.3.0 is directly connected, Loopback0

Rack1R1#show ip route | include via 155.1.13.3

Rack1R1#debug ip rip
RIP protocol debugging is on
RIP: ignored v2 update from bad source 155.1.13.3 on Serial0/1

Rack1R3#debug ip rip
RIP protocol debugging is on
RIP: received v2 update from 155.1.13.1 on Serial1/2
     155.1.146.0/24 via 0.0.0.0 in 1 hops
     192.10.1.0/24 via 0.0.0.0 in 3 hops
     204.12.1.0/24 via 0.0.0.0 in 3 hops
     205.90.31.0/24 via 0.0.0.0 in 10 hops
     212.18.1.0/24 via 0.0.0.0 in 6 hops
     212.18.3.0/24 via 0.0.0.0 in 6 hops
     220.20.3.0/24 via 0.0.0.0 in 10 hops
     222.22.2.0/24 via 0.0.0.0 in 10 hops
```

Once R1 is configured to ignore update source validation routes from R3 are installed.

```
Rack1R1#conf t
Enter configuration commands, one per line.  End with CNTL/Z.
Rack1R1(config)#router rip
Rack1R1(config-router)#no validate-update-source
Rack1R1(config-router)#end

Rack1R1#show ip route rip | include via 155.1.13.3
R       155.1.9.0/24 [120/3] via 155.1.13.3, 00:00:03
R       155.1.7.0/24 [120/2] via 155.1.13.3, 00:00:03
R       155.1.37.0/24 [120/1] via 155.1.13.3, 00:00:03
R       155.1.79.0/24 [120/2] via 155.1.13.3, 00:00:03
R       155.1.67.0/24 [120/2] via 155.1.13.3, 00:00:03
R       54.1.1.0 [120/3] via 155.1.13.3, 00:00:03
R    212.18.1.0/24 [120/4] via 155.1.13.3, 00:00:03
R    212.18.3.0/24 [120/4] via 155.1.13.3, 00:00:03
R       150.1.7.0 [120/2] via 155.1.13.3, 00:00:03
R       150.1.6.0 [120/3] via 155.1.13.3, 00:00:03
R       150.1.3.0 [120/1] via 155.1.13.3, 00:00:03
R       150.1.9.0 [120/3] via 155.1.13.3, 00:00:03
```